– EPE –
NEW NAME
NEW DESIGN!

Practical Electronics

The UK's premier electronics and computing maker magazine

Micromite
MMBASIC graphical commands

Electronic Building Blocks
Auto gadgets

Circuit Surgery
Transistor theory and practice

Cool Beans
Designing a 4-bit computer

ST57STH76-2804A
0800076026
130316008

Exciting new series on stepper motors

4 433 619Hz
MHz
kHz
OK
CANCEL
1
2
3
4
5
6
7
8
9
0
<-
GPS-SYNCHED PROGRAMMABLE FREQUENCY REFERENCE
TOUCH SCREEN FOR SETTINGS
SILICON CHIP

WIN!
Microchip PIC-IoT WG Development Board
WIN!

GPS-synced Frequency Reference

WIN!
$50 of PCB prototyping from PCBWay

SILICON CHIP
OPTO-ISOLATED RELAY
©2018

Opto-isolated Mains Relay

LS3/5A Crossover design

DCC Programmer for Decoders

ARDUINO

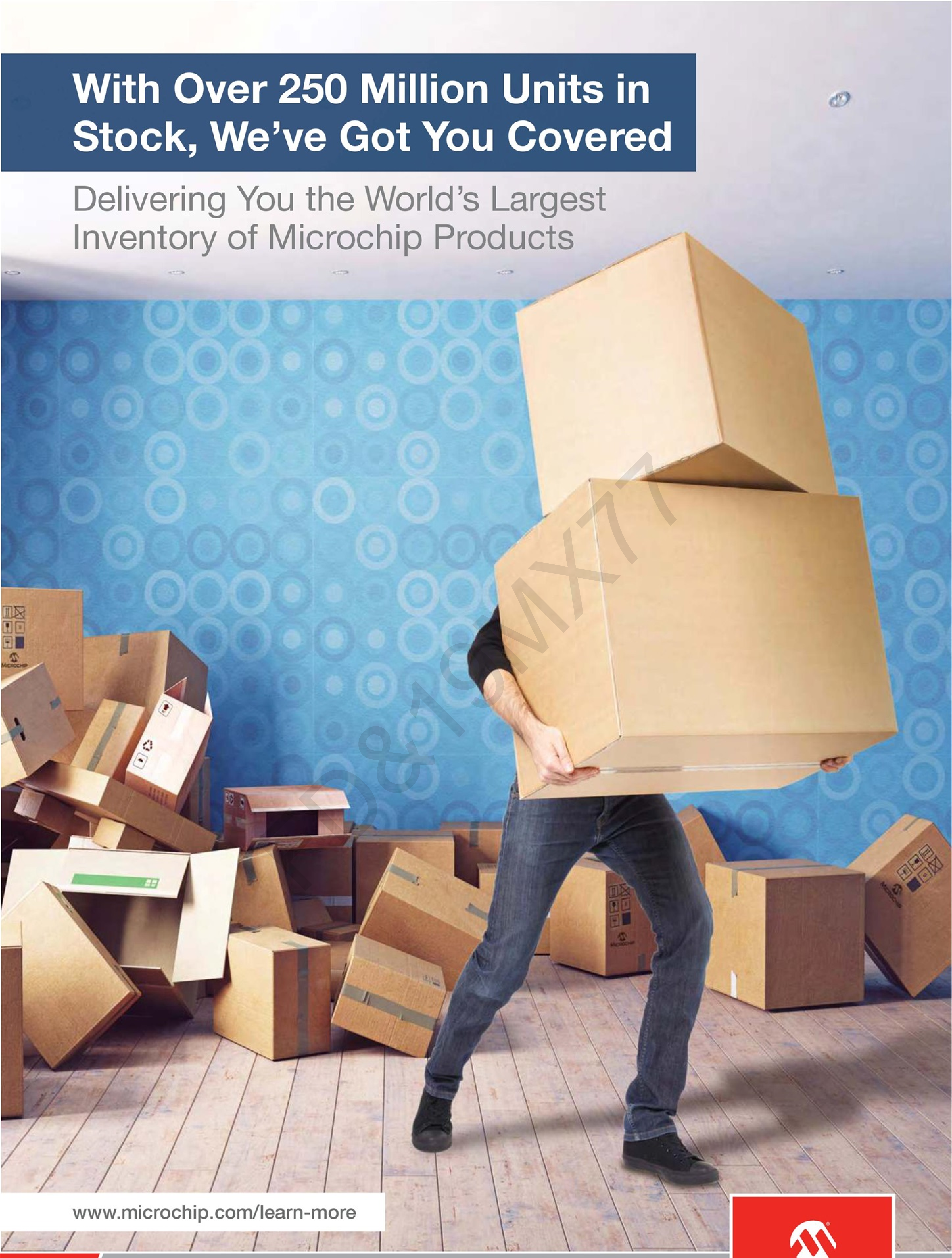
HELLO
0.96"80x160(RGB)IPS

See our Super Summer Sale!

PLUS!
Net Work – Look back to the start of the Internet
Techno Talk – Two cheers for 5G
The Fox Report – Finding free 4K content via satellite

Oct 2019 £4.65
10
9 772632 573009

www.epemag.com | @practicalelec | practicalelectronics



With Over 250 Million Units in Stock, We've Got You Covered

Delivering You the World's Largest Inventory of Microchip Products

2&19MX77

www.microchip.com/learn-more

MICROCHIP

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
All other trademarks are the property of their registered owners.
© 2018 Microchip Technology Inc. All rights reserved. DS00002552A. MEC2226Eng10/18

Practical
Electronics

Volume 48. No. 10
October 2019
ISSN 2632 573X

Contents

Projects and Circuits

Programmable GPS-synced Frequency Reference – Part 1 by Tim Blythman 14
For design, servicing, calibration or whenever you need a really accurate and stable frequency reference, this Micromite BackPack-based project delivers the goods.

Digital Command Control Programmer for Decoders by Tim Blythman 24
DCC is a widely-used method for controlling model railways. This programmer is simple, cheap and easy to build – and operates from a computer's USB port.

Opto-isolated Mains Relay by Tim Blythman 32
This project lets you switch the mains from a microcontroller or low-voltage output, but keeps dangerous voltages safely locked away.

Series, Features and Columns

The Fox Report by Barry Fox 8
Free 4K material via satellite

Techno Talk by Mark Nelson 10
Two cheers for 5G

Net Work by Alan Winstanley 12
A look back 50 years to the very start of the Internet...
Google's world of Cardboard... Augmented reality... Newsbeat

Using Stepper Motors by Paul Cooper 40
Part 1 – Introduction to stepper motor technology

Circuit Surgery by Ian Bell 44
Understanding bipolar junction transistors – Part 4

Audio Out by Jake Rothman 48
Designing Hi-Fi mini-monitor loudspeakers – Part 2

Make it with Micromite by Phil Boyce 58
Part 9: Introduction to MMBASIC graphical commands

Max's Cool Beans by Max The Magnificent 66
A little more on metastability, and back to the 'HRRG'

Electronic Building Blocks by Julian Edgar 76
Automotive gadgetry

Regulars and Services

PE Teach-In 8 3
Subscribe to Practical Electronics and save money 4
PE Summer Sale – a page full of great deals 6
Reader services – Editorial and Advertising Departments 7
Editorial 7
Competitions – please enter them!
Exclusive Microchip reader offer 23
Win a Microchip PIC-IoT WG Development Board
Exclusive PCBWay reader offer 38
Win one of six $50 credits to use with PCBWay's superb PCB prototyping service
Teach-In bundle – what a bargain! 56
Practical Electronics back issues CD-ROM – great 15-year deal! 57
Practical Electronics CD-ROMS for electronics 70
A superb range of CD-ROMs for hobbyists, students and engineers
Direct Book Service 73
Build your library of carefully chosen technical books
Practical Electronics PCB Service 78
PCBs for Practical Electronics projects
Classified ads and Advertiser index 79
Next month! – highlights of our next issue of Practical Electronics 80

© Electron Publishing Limited 2019

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The November 2019 issue of *Practical Electronics* will be published on Thursday, 7 October 2019 – see page 80.

Stepper motor image on cover and contents page courtesy of Pololu Robotics & Electronics, pololu.com

QUASAR
electronics

Quasar Electronics Limited
PO Box 6935, Bishops Stortford
CM23 4WP, United Kingdom

Tel: 01279 467799
E-mail: sales@quasarelectronics.co.uk
Web: quasarelectronics.co.uk

All prices INCLUDE 20.0% VAT. Free UK delivery on orders over £48
Postage & Packing Options (Up to 0.5Kg gross weight): **UK Standard 3-7 Day Delivery** - £4.95; **UK Mainland Next Day Delivery** - £9.95; **Europe (EU)** - £12.95; **Rest of World** - £14.95 (up to 0.5Kg).
!! Order online for reduced price postage and fast despatch !!
Payment: **We accept all major credit/debit cards. Make cheques/PO's payable to Quasar Electronics Limited.**
Please visit our online shop now for full details of over 1000 electronic kits, projects, modules and publications. Discounts for bulk quantities.

QuasarElectronics.co.uk

Solutions for Home, Education & Industry Since 1993

cebek
velleman

Official Main Dealer stocking the full range of Kits, Modules, Robots, Instruments, tools and much, much more...

PIC & ATMEL Programmers

We have a wide range of PIC, ATMEL Arduino and Raspberry Pi projects.

PIC Programmer & Experimenter Board

Great learning tool. Includes programming examples and a reprogrammable 16F627 Flash Microcontroller. Test buttons & LED indicators. Software to compile & program your source code is included. Supply: 12-15Vdc. Pre-assembled and ready to use.
Order Code: VM111 - ***~~£38.88~~ £35.94***

USB PIC Programmer and Tutor Board

The only tutorial project board you need to take your first steps into Microchip PIC programming using a PIC16F882 (included). Later you can use it for more advanced programming. Programs all the devices a Microchip PICKIT2® can! Use the free Microchip tools for PICKit2™ & MPLAB® IDE environment.
Order Code: EDU10 - ***£46.74***

USB /Serial Port PIC Programmer

Fast programming. Wide range of PICs supported (see website for details). Free Windows software & ICSP header cable. USB or Serial connection. ZIF Socket, leads, PSU not included.
Kit Order Code: 3149EKT - ***~~£49.96~~ £29.95***
Assembled Order Code: AS3149E - ***£44.95***
Assembled with ZIF socket Order Code: AS3149EZIF - ***~~£74.96~~ £49.95***

PICKit™2 USB PIC Programmer Module

Versatile, low cost, PICKit™2 Development Programmer. Programs all the devices a Microchip PICKIT2 programmer can. Onboard sockets & ICSP header. USB powered.
Assembled Order Code: VM203 - ***£35.94***

Controllers & Loggers

Here are just a few of the controller and data acquisition and control units we have. See website for full details. 12Vdc PSU for all units: Order Code 660.446UK £10.68

USB Experiment Interface Board

Updated Version! 5 digital inputs, 8 digital outputs plus two analogue inputs and two analogue outputs. 8 bit resolution. DLL.
Kit Order Code: K8055N - ***~~£39.95~~ £22.20***
Assembled Order Code: VM110N - ***£35.94***

2-Channel High Current UHF RC Set

State-of-the-art high security. Momentary or latching relay outputs rated to switch up to 240Vac @ 12 Amps. Range up to 40m. 15 Tx's can be learnt by one Rx. Kit includes one Tx (more available separately). 9-15Vdc.
Kit Order Code: 8157KT - ***£44.95***
Assembled Order Code: AS8157 - ***£49.96***

Computer Temperature Data Logger

Serial port 4-ch temperature logger. °C/°F. Continuously log up to 4 sensors located 200m+ from board. Choice of free software applications downloads for storing/using data. PCB just 45x45mm. Powered by PC. Includes one DS18S20 sensor.
Kit Order Code: 3145KT - ***~~£19.95~~ £16.97***
Assembled Order Code: AS3145 - ***£19.96***
Additional DS18S20 Sensors - ***£4.96 each***

8-Channel Ethernet Relay Card Module

Connect to your router with standard network cable. Operate the 8 relays or check the status of input from anywhere in world. Use almost any internet browser, even mobile devices. Email status reports, programmable timers... Test software & DLL online.
Assembled Order Code: VM201 - ***£130.80***

Computer Controlled / Standalone Unipolar Stepper Motor Driver

Drives any 5-35Vdc 5, 6 or 8-lead unipolar stepper motor rated up to 6 Amps. Provides speed and direction control. Operates in stand-alone or PC-controlled mode for CNC use. Connect up to six boards to a single parallel port. Board supply: 9Vdc. PCB: 80x50mm.
Kit Order Code: 3179KT - ***£15.26***
Assembled Order Code: AS3179 - ***£22.26***

Card Sales & Enquiries 01279 467799

Bidirectional DC Motor Speed Controller

Control the speed of most common DC motors (rated up to 32Vdc/5A) in both the forward and reverse directions. The range of control is from fully OFF to fully ON in both directions. The direction and speed are controlled using a single potentiometer. Screw terminal block for connections. PCB: 90x42mm.
Kit Order Code: 3166KT - ***£19.99***
Assembled Order Code: AS3166 - ***£29.99***

8-Ch Serial Port Isolated I/O Relay Module

Computer controlled 8 channel relay board. 5A mains rated relay outputs and 4 opto-isolated digital inputs (for monitoring switch states, etc). Useful in a variety of control and sensing applications. Programmed via serial port (use our free Windows interface, terminal emulator or batch files). Serial cable can be up to 35m long. Includes plastic case 130x100x30mm. Power: 12Vdc, 500mA.
Kit Order Code: 3108KT - ***£74.95***
Assembled Order Code: AS3108 - ***£89.95***

8-Channel RF Remote Control Set

Control 8 onboard relays with included RF remote control unit. Toggle or momentary mode for each output. Up to 50m range. Board Supply: 12Vac, 500mA
Assembled Order Code: VM118 - ***£71.94***

Temperature Monitor & Relay Controller

Computer serial port temperature monitor & relay controller. Accepts up to four Dallas DS18S20 / DS18B20 digital thermometer sensors (1 included). Four relay outputs are independent of the sensors giving flexibility to setup the linkage any way you choose. Commands for reading temperature / controlling relays are simple text strings sent using a simple terminal or coms program (e.g. HyperTerminal) or our free Windows application. Supply: 12Vdc.
Kit Order Code: 3190KT - ***~~£79.96~~ £47.95***
Assembled Order Code: AS3190 - ***£59.95***

3x5Amp RGB LED Controller with RS232

3 independent high power channels. Preprogrammed or user-editable light sequences. Standalone or 2-wire serial interface for microcontroller or PC communication with simple command set. Suits common anode RGB LED strips, LEDs, incandescent bulbs. 12A total max. Supply: 12Vdc. 69x56x18mm
Kit Order Code: 8191KT - ***£24.95***
Assembled Order Code: AS8191 - ***£27.95***

Soundtronics.co.uk

Build your dream modular analogue synth from our range of over 80 kits, including front panels, components and PCBs

Components for the synth builder and hobbyist
- Resistors and Capacitors
- Diodes, Transistors and ICs
- Switches and Potentiometers
- Knobs, sockets, connectors, leads and more

Order online at
www.soundtronics.co.uk
Unit 29, Webb Ellis Business Park
Rugby, Warwickshire
CV21 2NP
Tel: 01788 878 002

Soundtronics MIDI Ultimate Analogue Synth

As featured in Practical Electronics February – July 2019. Available as PCB, components, case or full kits

- Three VCOs
- Mixer
- 12dB VCF
- Two LFOs
- VCA
- Two ADSRs
- Sample & Hold
- Noise Source
- Repeat Gate Generator
- Echo Effect
- MIDI Interface
- Headphone Amplifier
- Toggle Switch Patching

Teach-In 8 CD-ROM
Exploring the Arduino

This CD-ROM version of the exciting and popular *Teach-In 8* series has been designed for electronics enthusiasts who want to get to grips with the inexpensive, immensely popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. *Teach-In 8* provides a one-stop source of ideas and practical information.

The Arduino offers a remarkably effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle wirelessly or via the Internet. *Teach-In 8* is based around a series of practical projects with plenty of information for customisation. The projects can be combined together in many different ways in order to build more complex systems that can be used to solve a wide variety of home automation and environmental monitoring problems. The series includes topics such as RF technology, wireless networking and remote web access.

FREE CD-ROM
£8.99
ELECTRONICS TEACH-IN 8
FROM THE PUBLISHERS OF EPE EVERYDAY PRACTICAL ELECTRONICS
FREE CD-ROM SOFTWARE FOR THE TEACH-IN 8 SERIES
INTRODUCING THE ARDUINO
• Hardware – learn about components and circuits
• Programming – powerful integrated development system
• Microcontrollers – understand control operations
• Communications – connect to PCs and other Arduinos
Teach-In 8
Exploring the Arduino
PLUS...
PIC n'MIX
PICs and the PICkit 3 - A beginners guide. The why and how to build PIC-based projects

PRICE
£8.99
Includes P&P to UK if ordered direct from us

PLUS: PICs and the PICkit 3 – A beginners guide
The CD-ROM also includes a bonus – an extra 12-part series based around the popular PIC microcontroller, explaining how to build PIC-based systems.

SOFTWARE
The CD-ROM contains the software for both the *Teach-In 8* and PICkit 3 series.

ORDER YOUR COPY TODAY!
JUST CALL 01202 880299 OR VISIT www.epemag.com

– EPE – NEW NAME NEW DESIGN!

Practical Electronics

UK readers SAVE 73p on every issue

Why not pay £3.92 instead of £4.65 for your copy of Practical Electronics?

Just take out a one-year subscription and save 73p on every issue, or £8.80 over the year. Even better – save 94p per issue if you subscribe for two years – a total saving of £22.60

Subscribe today and avoid the price rise on 1 October 2019 – see page 6 for special offers!

FD819MX77

Practical Electronics

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

UK: 6-months £25.00, 12-months £47.00, 24-months £89.00

Europe Airmail: 6-months £30.00, 12-months £56.00, 24-months £107.00

Rest Of The World Airmail: 6-months £37.00, 12-months £70.00, 24-months £135.00

Cheques (in **£ sterling only**) payable to Practical Electronics and sent to: Wimborne Publishing Ltd, 113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU.

Tel: 01202 880299. **Fax:** 01202 843233.
Email: stewart.kearn@wimborne.co.uk.

Also via our secure online shop at**: www.epemag.com**.

Subscriptions start with the next available issue. We accept MasterCard or Visa.
(For past issues see the Back Issues page.)

ONLINE SUBSCRIPTIONS

Electronic subscriptions (PDF download or online) are available, £22.00 for 12 months, visit **www.epemag.com** for more details.

SUBSCRIPTION ORDER FORM

☐ 6 Months: UK £25.00, Europe £30.00 (Airmail), Rest of the World £37.00 (Airmail)

☐ 1 Year: UK £47.00, Europe £56.00 (Airmail), Rest of the World £70.00 (Airmail)

☐ 2 Years: UK £89.00, Europe £107.00 (Airmail), Rest of the World £135.00 (Airmail)

To: Practical Electronics
Wimborne Publishing Ltd
113 Lynwood Drive, Wimborne, Dorset BH21 1UU

Tel: 01202 880299 Fax: 01202 843233
Email: stewart.kearn@wimborne.co.uk

I enclose payment of £ ....
(cheque/PO in £ sterling only)
payable to *Practical Electronics*

☐ Please charge my Visa/Mastercard

My card number is: ....
Please print clearly, and check that you have the number correct

Card Security Code .... Valid From Date.......................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ....

Name ....

Address ....

Post code .... Tel. ....

Email ....

Subscriptions start with the next available issue.

Console
steam
beans
milk
Coffee machine

Your Choice—Any Core, Any Performance, Any Feature Set

Scalable Performance for Changing Requirements

Are you facing changing design requirements–again? Let Microchip help you put an end to the frustrations and wasted time that result from these changes. Microchip is the only semiconductor supplier innovating across 8-, 16- and 32-bit microcontrollers, digital signal controllers and microprocessors. Our upward-compatible architectures preserve your time and resource investment in code development. In addition, our development ecosystem allows you to leverage a common ecosystem across multiple designs. Changing design requirements don't need to be painful; learn how Microchip can make it effortless.

Simplify your life at www.microchip.com/Scalable

MICROCHIP

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2019 Microchip Technology Inc. All rights reserved. DS30010186A. MEC2290A-ENG-07-19

Super Summer Sale!

Subscribe now and avoid October's price rise!

Practical Electronics special offers

40% off all PCBs up to and including those in the February 2019 issue
25% off all PE/EPE hard-copy (print) back issues
25% off all PE/EPE back-issue 6-month CD-ROMs – see page 57
25% off all PE/EPE back-issue 5-year CD-ROMs – see page 57

Practical Electronics subscription offer

Subscribe to PE hard copy for 12-months and receive a **free 6-month back-issue CD-ROM** of your choice from Volume 1-40 (see page 57), normal price £16.45.

Subscribe to PE hard copy for 24-months and receive **three free 6-month back-issue CD-ROMs** of your choice from Volume 1-40 (see page 57), normal price £16.45 each.

Existing subscribers are welcome to renew early for another 12 or 24 months and still receive the offer – your new subscription will simply be added to the end of your existing subscription – **no issues lost by renewing early and you avoid October's subscription price rise!**

FD&19M/47

Teach-In bundle

Electronics Teach-In bundle includes
TI CD-ROMs 3, 4 and 5 – normal price £18.95
Special offer 25% discount price £14.21 – see page 56

PAYMENT MUST BE RECEIVED BY CLOSE
OF OFFER: 30 SEPTEMBER 2019

Don't miss out!
Just call 01202 880299 or visit our secure online shop at:
WWW.EPEMAG.COM

Practical Electronics

Volume 48. No. 10
October 2019
ISSN 2632 573X

Editorial

Editorial offices
Practical Electronics
Electron Publishing Limited
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Tel 01273 777619
Mob 07973 518682
Fax 01202 843233
Email pe@electronpublishing.com
Web www.epemag.com

Advertisement offices
Practical Electronics Adverts
1 Buckingham Road
Brighton
East Sussex BN1 3RA

Phone 01273 777619
Mob 07973 518682
Email pe@electronpublishing.com

Editor Matt Pulzer
General Manager Louisa Pulzer
Subscriptions Stewart Kearn
Online Editor Alan Winstanley
Web Systems Kris Thain
Publisher Matt Pulzer

Technical enquiries
We regret technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine. We cannot provide data or answer queries on articles or projects that are more than five years old.

Questions about articles or projects should be sent to the editor by email: pe@electronpublishing.com

Projects and circuits
All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

A number of projects and circuits published in *Practical Electronics* employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

Component supplies
We do not supply electronic components or kits for building the projects featured, these can be supplied by advertisers. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

Advertisements
Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

Transmitters/bugs/telephone equipment
We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

Competitions – please enter them!

First, some good news. I'm pleased to tell you that we have a new competition this month (and next). Our recommended PCB provider – PCBWay, as used by *PIC n' Mix's* Mike Hibbett and *Audio Out's* Jake Rothman – are generously offering six $50 prizes. All you have to do is use the link on page 38 to set up an account, and six lucky readers will be drawn at random on 1 October. Setting up an account is free and easy – you have nothing to lose and may well win a nice prize.

This is in addition to our regular competition from Microchip (page 23), who have generously supported the magazine for many years. Please do enter both company's competitions; it really does tell them that supporting *Practical Electronics* is worthwhile – thank you!

Price rise

And now some less welcome news… it's been nearly two and a half years since the last magazine price rise, and unfortunately increases in paper and distribution costs, not to mention inflation mean that we need to raise the cover price of *Practical Electronics* to £4.99 from the next issue (November). Subscriptions will rise at the same rate (approximately 7.5%).

We really do try and keep our costs to a minimum, but we also need to charge a fair price that ensures the future of *Practical Electronics*.

With that in mind, let me make one final reminder that we're still running our annual summer sale (see opposite). If you renew your subscription between now and the end of September then not only will you receive your choice of CD-ROM(s), but also you're guaranteed to avoid the price rise. So, subscribe by the end of September for 24 months and you'll avoid the price rise until the summer of 2021.

Remember, even if your subscription is not due yet, you won't lose out; your additional subscription will simply start after the current one ends and you're still guaranteed to avoid any price rises.

What are you waiting for? Grab a bargain!

Matt Pulzer
Publisher

Free 4K material via satellite

By looking outside the usual SES/Astra satellite broadcasts of UK Freesat and free Sky programmes, Barry Fox has discovered additional no-charge 4K HDR material.

So, you have your nice new 4K UHD TV, but all you can watch on it is 2K material. You don't want to buy a new Blu-ray player and UHD discs, or pay Amazon or Netflix for streamed 4K, which your broadband connection may or may not be able to support.

Exchanging 4K home-grown movie files with friends is an 'orrible obstacle course to navigate.

I recently failed so many times to download a friend's two-hour item in 4K, which soaked up over 60 GB, that I finally got on a train to collect a physical copy on an SSD HDD. In fact, I had to do the journey twice because I had not realised that my 4K TV would only read from an NTFS-formatted disc (not exFAT).

Free 4K channels

However, I then discovered a free source of 4K HDR material, broadcast direct to home by satellite. The content is travel and lifestyle, not drama. But beggars can't be choosers.

The trick is to look outside the bundle of UK Freesat and free Sky programmes broadcast from the SES/Astra satellite(s) at 28.2 East. I discovered this by accident while trying to use a new Philips 4K HDR TV with a satellite dish, rather than a normal Freeview terrestrial TV aerial, because Freeview reception on the UK's South Coast is bedevilled by interference from French stations in warm weather.

Accessing new channels

Although details may vary, the basic method of tuning is likely to be the same for other pan-European 4K TVs that come with a satellite socket as well as a terrestrial aerial socket. I did the following:

- Connect the set's satellite socket to a single coax feed from any working Sky or Freesat dish.
- Go to: Settings > Install Channels, and choose Satellite Installation. Ignore any offer to use any Freesat software that the set may have and select a generic satellite option.
- Then click to 'Search for Satellite' and the set will find the 28.2 East bird(s) because that is the only satellite slot a correctly installed Sky/Freesat dish will be 'seeing'.
- Click 'Install Channels' – and the set will find around 1000 TV and radio channels.
- Go to List or similar option on the remote control, choose Satellite TV, and the TV will show all the channels found, some with unhelpful numbering and gobbledegook names, but most with helpful labels such BBC News HD, BBC Four HD or Sky News HD.
- Avoid anything marked with a key icon, which denotes 'scrambled'. These are for use by Sky boxes with integrated de-encryption or TVs with European Conditional Access modules plugged in.
- Press whichever button on the remote control offers to mark chosen channels as Favourites. On the Philips set it is the green button.
- From then on, use the Satellite Favourites channel List to choose channels to watch.

In among all the familiar Freesat and free Sky channels you should find several dedicated to 4K HDR demo channels, including a couple from Astra/SES and one intriguingly labelled 'BBC Olympics'. This has presumably been set aside for the 2020 Tokyo Games. Who knows, it may well carry some 4K feeds.

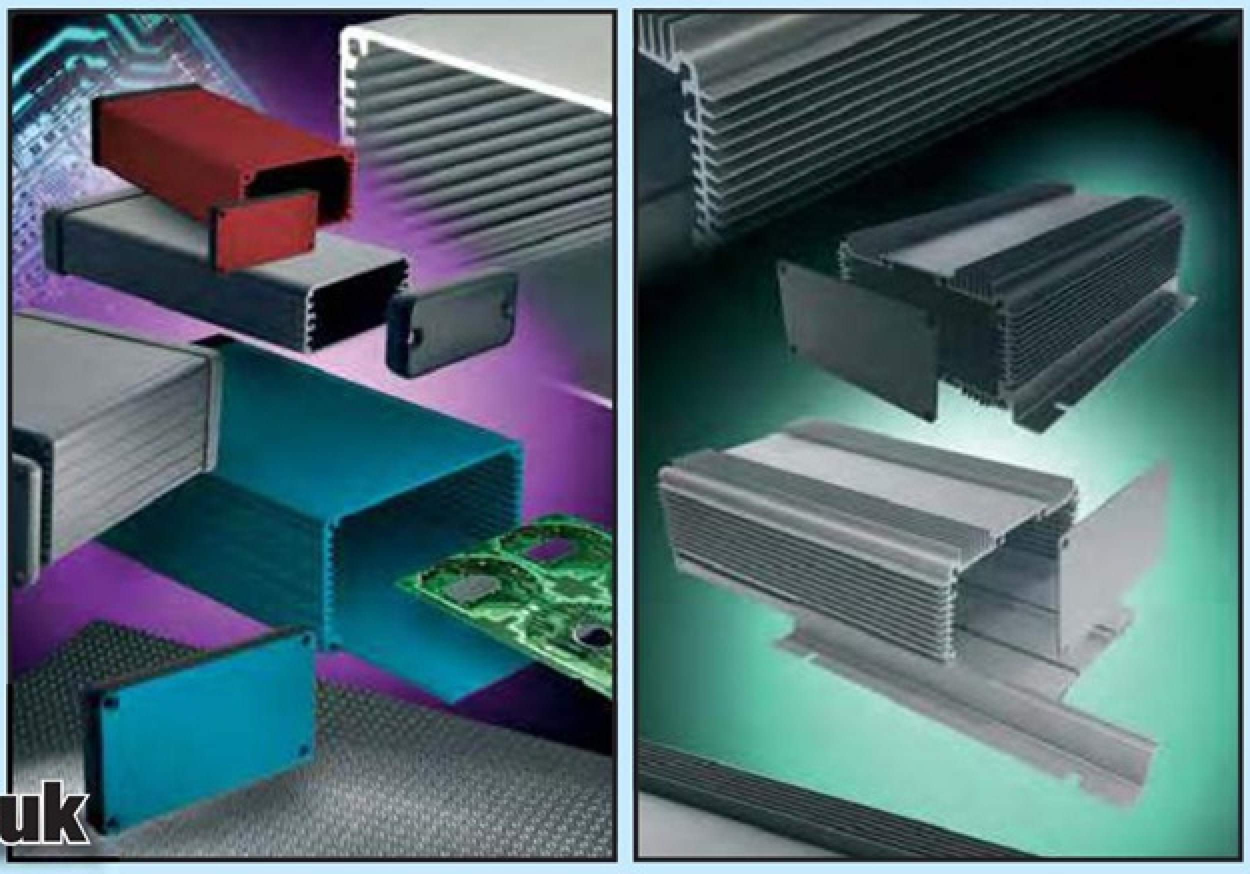
FD&19MX77

HAMMOND MANUFACTURING®

Extruded enclosures standard and heatsink

www.hammondmfg.com/1455.htm
www.hammondmfg.com/1455NHD.htm

01256 812812
sales@hammond-electronics.co.uk



PCB Layout and Design

A comprehensive range of electronic design and development services to a number of international clients. Through our professional design team, our clients benefit from a cost-effective, fully integrated electronic solution for the complete development of their products.

Student & Non-profit Project Sponsorship

Free PCB prototypes or special discount to bring your ideas to life with PCBWay support from prototype to production run.

PCB Prototype and Production

The online pricing system can instantly quote for your PCB manufacturing service from us, though a rough calculation, you can still have some idea of the real cost which helps you decide to use our service. Not only the standard boards, but also advanced PCBs are avaliable - Rogers,Flexible, Flex-Rigid, Aluminum and HDI boards.

Open Source Community

Platform where people can share their open source projects with others.

PCBWay

SMT & THT Assembly

PCBWay has accumulated lots of reliable and competitive parts suppliers around the world to ensure the quality and cost of your parts sourcing. All your jobs can be done under one roof, either SMT, Thru-hole or Hybrid!

24/7 Customer Support Service

Round-the-clock service to solve your order problem quickly and timely

Get an instant quote online: www.pcbway.com 24/7 customer service: service@pcbway.com

Two cheers for 5G

Techno Talk

Mark Nelson

You've probably read the good news about 5G – the fabulous new fifth-gen mobile comms and data technology, packaged as a digital blessing for mankind. However, new technology has a nasty habit of springing surprises and biting where it hurts, which is why an increasing number of informed observers are voicing concerns over 5G.

But first, what's not to like about 5G? It promises considerably higher data rates, lower latency (time-lag) plus an increased number of simultaneous connections. What's the problem? In a word, 'health risk'. (That's two words, but never mind.)

Cast your mind back 110 years to the early days of research into radioactivity. Although Marie and Pierre Curie were keen proponents of the opportunities offered by their discoveries of polonium and radium, they had absolutely no idea about the damaging effects of ionising radiation. Marie carried test tubes containing radioactive isotopes in her pocket, and was also exposed to X-rays from unshielded equipment while serving as a radiologist in First World War field hospitals. Her husband was equally unaware of the health risks of radiation exposure; experimenting with radium next to the skin of his arm.

Ionising radiation

Radiation mirrors the possible problem with 5G that many argue has not been examined adequately. As the authoritative website **EETech.com** observes, the effect of radio waves on health has been a matter of concern for decades. The onset of 5G has revived these concerns, as new hardware presents fresh opportunities for misunderstandings, not to mention the need for better education. Many experts, it acknowledges, do dispute the idea of any health-related concerns, because mobile-technology radio waves are in the centimetre range, and therefore *non*-ionising (they can cause heating, though). 5G, however, uses much higher frequencies, which are millimetric and thus that much closer to the ionising range. Add to this an increased number of base stations, and there are those who question the technology's safety.

Ionising radiation, it should be explained, is radiation that carries sufficient energy to detach electrons from atoms or molecules – thereby ionising them. It is harmful and potentially lethal, causing radiation burns, cancer and heritable effects. The boundary between ionising and non-ionising electromagnetic radiation is not sharply defined: different molecules and atoms ionise at differing energy levels. Overall, some argue far more study needs to be made on the long-term effects of exposure to millimetre radiation.

Bee-stly effects

Of course, it's not just us humans who may be affected. Our loyal friends the bees, which act as vital pollinators for our food production, are already under attack by neonicotinoid insecticides, colony collapse disorder and verroa mite disease. In this situation, 5G radio could become a further threat. At the 26GHz frequency band allocated for 5G in the UK and Europe, the wavelength (11.5mm) is close to the bee's size and radiation penetrates further into the body.

A campaigning organisation, *Physicians for Safe Technology* (**http://bit.ly/pe-oct19-tt**), cites academic reports that bee behaviour is already being disrupted by exposure to cell phone GSM radiofrequency radiation, causing worker bees to emit a piping signal to swarm.

Industry denial

The foregoing may sound contentious, but there is not sufficient scientific evidence to prove one way or another. The usual industry excuse is that we have been bombarded by radiation from broadcast transmitters for many decades and from cellular towers for the past 25 years, with no conclusive correlation between exposure to radio frequency energy and health. This is specious, because that radiation was at a significantly lower frequency.

To underscore this, campaigner Jeffery Naydler, states: 'Earlier this year, *The Guardian* published an article citing research that showed that while 67% of independently funded studies found a biological effect of exposure to radio-frequency radiation, only 28% of industry-funded studies did. Industry-funded studies are almost two-and-a-half-times less likely than independent studies to find health effects. One of the reasons for ignoring the evidence, in the hell-for-leather dash to create the 5G electronic ecosystem, is the conviction in government circles that, unless we introduce it immediately, we will be left behind and our economic growth and competitiveness will be put at risk. There is simply no time to consider the possible health consequences.'

To hell in a handcart?

Are we, as some observers claim, rushing headlong into a new nirvana of total connectivity? Jeffrey Naydler further argues that, 'the electromagnetic energies that we have summoned, and which promised to give us new powers, now appear to be overpowering us, carrying us down into an Underworld realm of hellish entertainment and distraction, of illusion and disconnection from the reality that really matters – the reality of nature and the spiritual order that infuses nature, the reality of fellow creatures with whom we share our world, living in soil and sky and walking the Earth beside us.'

He suggests we demand greater scrutiny of 5G's rollout – and protest. There are campaigns we can join, petitions to sign, letters to write and legal actions to support. The best place to begin is the *International Appeal to Stop 5G on Earth* and in Space (**www.5gspaceappeal.org**), which has already gathered more than 50,000 signatures, including those of many health professionals, environmentalists and other scientists.

More speed less haste

Smoke for a day and you'll probably be fine (but please don't risk it), smoke for 30 years and you massively increase the risk of fatal disease. Where and how do you draw the line? Perhaps the real point with 5G is we just don't know. It offers real and important benefits, but as the frequencies and power levels climb so should our vigilance. 5G signals will be everywhere all the time, so even if they turn out to be relatively harmless, it is well worth understanding the effect of 24/7 exposure to high (and rising) frequency non-ionising radiation on not just us, but the ecosystems we depend on.

FD&19NX77

pico Technology

TC-08: Your 8-Channel Thermocouple Data Logger

Take your first temperature logger measurements in three simple steps

1 Connect

2 Configure

3 Collect

TC-08
23.27 °C
pico
Technology

When paired with the power and astounding flexibility of PicoLog 6 software, the TC-08 temperature logger can start recording reliable, accurate, detailed measurements in under 60 seconds.

Only £289 (excl VAT)

To find out more visit: **www.picotech.com/A396**

Net Work

Alan Winstanley

This month, we look back 50 years to the very start of what became the Internet, as a pair of computers 350 miles apart in California formed a simple network and exchanged data.

Next month marks a milestone for the Internet as we know it today: precisely 50 years ago the very first attempt was made to log into a remote computer over a network. This momentous achievement is attributed to Leonard Kleinrock, an American engineer who devoted the early part of his career to researching principles of transmitting data over networks, an idea that attracted very little interest back in the early 1960s.

Like many of the best engineers, Leonard Kleinrock started out as an electronics hobbyist who was fascinated by early crystal radios and the inner workings of radio sets. He honed his electronic engineering skills at science school before entering university. At the time, the emerging science of information technology was somewhat over-subscribed, Kleinrock felt, and so he decided to focus his attention on something brand new: data networks.

It is said that in a paper he wrote back in 1962, Leonard Kleinrock first floated the basic idea of splitting large chunks of data into smaller pieces that were more suited for transmission over a network. Research on transmitting data using 'message blocks' was also carried out by Paul Baran of America's RAND Corporation. Their work coincided with that of Donald Davies of Britain's National Physical Laboratory (NPL), though Davies is generally credited with devising the core principles of 'packet switching', as NPL called it, in 1965. The NPL also demonstrated to the US Advanced Research Projects Agency (ARPA) the benefits of running dedicated high-speed data networks. In the new Sputnik era, the defence department had been charged with a building a resilient communications network – the ARPANET – to counter threats posed by the emerging satellite and nuclear age at a time when American telecoms relied on slow and busy phone lines. ARPA was soon sold on the new idea of using packet switching at the core of a 'digital' network.

Packet switching and the UK's NPL

According to archive footage, it is claimed that the NPL's own research on packet switching influenced the design of the first network hardware created by Bolt, Beranek and Newman in America in 1969, when they produced the world's first data switch – the IMP or Interface Message Processor – based on a rack of Honeywell minicomputers. In August 1969 the first IMP was delivered to Leonard Kleinrock at UCLA (University of California at Los Angeles) where he and his ARPA-sponsored team of engineers aimed to connect the switch to its first 'host' computer, a technical challenge that had never been attempted before. Interest in this event was industry-wide and, happily, data successfully flowed between the IMP and the host for the first time on 2 September 1969, a day Kleinrock dubbed 'the day the infant Internet took its first breath'. (See *Personal History/ Biography: the birth of the Internet*, at: **www.lk.cs.ucla.edu/personal_history.html**)

IMP – the world's first data switch – towering over Dr Leonard Kleinrock, 'the grandfather of the Internet'. (Image: UCLA).

Network

A second node was then installed over at Stanford Research Institute (SRI, 350 miles to the north in what was about to become Silicon Valley), and towards the end of October UCLA endeavoured to connect to it over a remote network, another ground-breaking event. Notable too was the early use of autocomplete: after sending the first few characters 'l-o-g' the SRI host was set up to fill the field with 'login', but after receiving the 'g' the host crashed! The remote transmission of the first successful login characters took place fifty years ago on 29 October 1969, the day 'the infant Internet uttered its first words'. Two months later, a total of four US sites had been hooked together on the embryonic ARPANET. Although only a dozen or so nodes were online by the mid 1970s, it was enough to develop, debug, refine and expand a packet-switching network that was the precursor to today's Internet.

The NPL's own story of packet switching can be found on YouTube at: **https://youtu.be/tT4AaelwvV4**, a fascinating documentary and a 'must see' for *Net Work* readers. It explains the gradual development of the Internet proper – an inter-networking data communications system that eventually interfaced with a computer network overseas, all reliant on the principle of packet switching. As they explain, the irony is that thanks to VoIP (Voice over Internet Protocol), data networks have largely taken over the role of those old, slow analogue lines that once formed the indispensable backbone of voice communications.

A Cardboard world

More than three years ago in *Net Work* (June 2016) I trialled a Google-Cardboard-compatible viewer, a simple but effective VR (virtual reality) device that

FD&19MX77

was made from diecut E-flute corrugated cardboard. It holds a smartphone in place and displays a stereoscopic split-screen of 'Cardboard compatible' media for viewing VR video through a pair of glass lenses. The viewer opens a new world of virtual reality, where compatible apps and videos can be viewed on-screen in an immersive 360° experience. Google Cardboard and a range of compatible 'Works with Cardboard' viewers from other suppliers can be sourced via **http://bit.ly/nw-oct19-360** or eBay, typically starting from just £2.99. Decent-quality VR visors made of plastic can be sourced from the usual online suppliers for next to nothing; unlike Cardboard, they may feature adjustable inter-pupillary distance (IPD) and focal length settings, making them a good choice for spectacle wearers. The 'VR Box' visor used by the author is a very inexpensive entrée into VR and 360° video, and is a good choice for those wishing to dabble for the first time.

The free Cardboard app for Android and iOS is available from Play Store. Paid-for media is also sold, such as a VR tour of London's famous Abbey Road studios, or a spaceman's view of the world (Astronaut VR), which the author found was a 'fast game played slow', but it might appeal to youngsters. YouTube also carries free Cardboard-compatible videos that often seem to consist of roller coaster rides or (be warned) house of horror videos. For family friendly viewing, a good place to start is by opening the '360 video channel' in the Cardboard app, and remember to find and click the 'Cardboard' icon to open the video in split screen before popping the smartphone into your VR viewer. As one example, motorsport fans might enjoy being a passenger while lapping the Nürburgring circuit in a monster Jaguar XE SV, complete with audio, and thanks to the 360° panoramic video you can even nose around the car too. It's great fun!

Augmented reality

Apart from virtual reality, a number of popular augmented reality (AR) apps are available that make the most of your smartphone or tablet. If your budget doesn't stretch to a GPS computer-controlled telescope, then one astronomy app to try is SkyView by Terminal Eleven. This AR program uses GPS to determine location (or enter yours manually) and displays the night sky on screen. Constellations and some satellites are recognised and superimposed on the screen as well, and individual stars and planets can be pinpointed by the cursor as the phone is panned around, 360° panorama style. Outdoors, the phone's camera can be used to point at the night sky or an AR scene of the night sky can be rendered indoors instead. A moody musical score adds to the experience and, overall, SkyView provides an enchanting view of some of the wonders of our night skies. Android and iOS versions are available, including a free 'Lite' version to get you started. A handful of Play Store reviewers are less than kind considering the low cost, but it's worth a try and is just the thing to spark an interest in astronomy as the nights start to draw in.

A 360° VR film of a Jaguar sports car on Nürburgring, available for Cardboard and compatible VR visors.

Newsbeat

Back to cardboard tech – Kodak has produced a cheap film scanner. Utilising the Kodak Mobile Film Scanner app, the clever design uses rigid cardboard components coupled with an LED backlight that enables a smartphone to scan 35mm positive and negative film strips and slides. It collapses into a 6-inch box and it might be all that some users need to scan a few old films. Presently it is only available in the US.

France is considering ways of protecting its satellites orbiting in space. As part of a new weapons program called *Maîtrise de l'Espace* (Masters of Space), proposals include shielding military satellites with swarms of 'nano satellites' or equipping them with high-power lasers.

Kodak Mobile Film Scanner
Kodak's card/smartphone film scanner

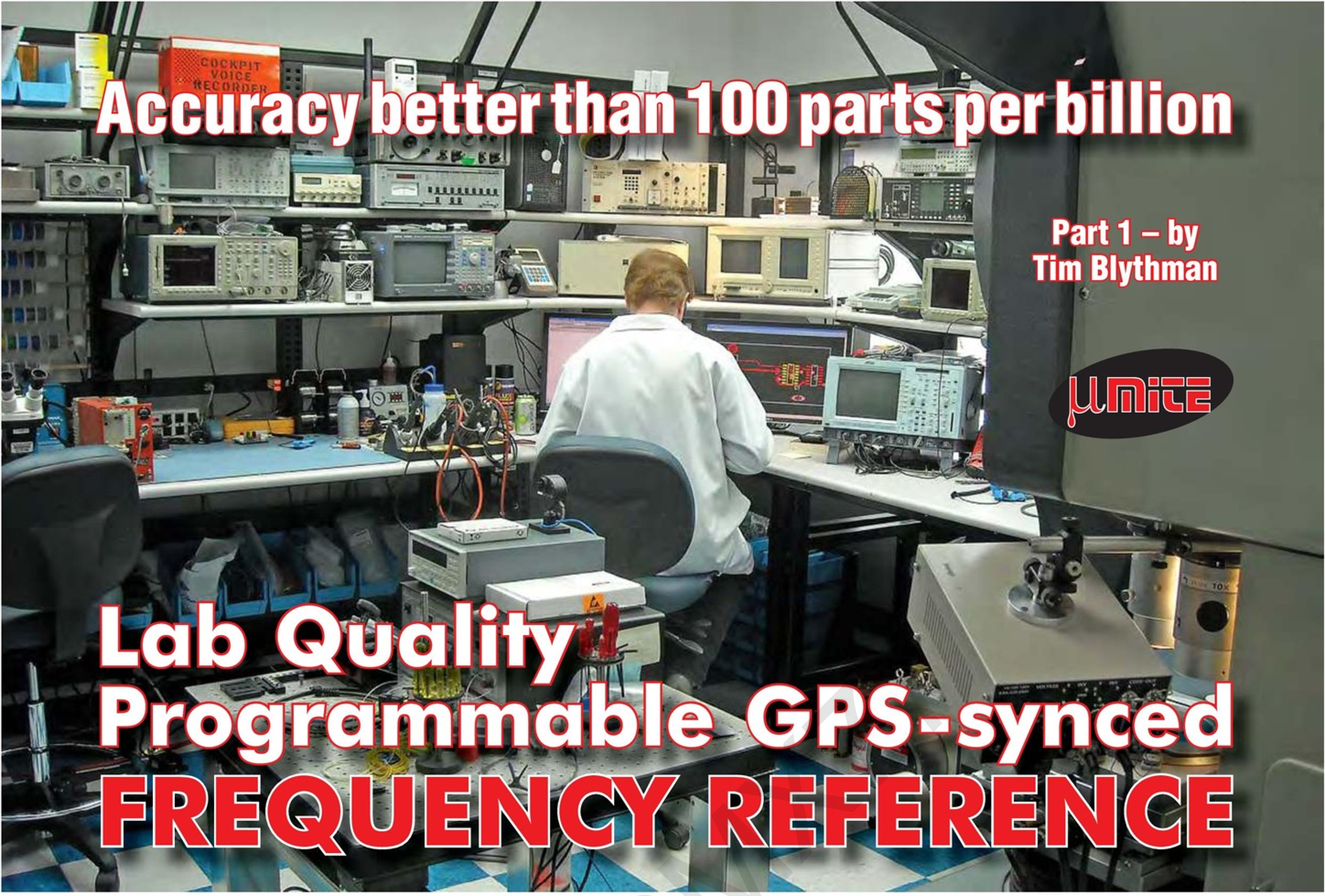
Back on Earth, France has implemented a new 3% digital sales tax aimed at raising tax revenue on 'soft' trade generated by tech giants such as Facebook and Google. European efforts to implement a digital sales tax (see *Net Work*, January 2019) fell through due to a lack of universal agreement among EU member states. All eyes are now on the OECD tackling the issue of global taxation in an Internet-enabled market. The UK is applying a 2% digital levy of its own.

Good news for Microsoft Windows stick-in-the-muds: it appears that rounded corners on applets and windows will be reintroduced in a move away from the stark, barren squared-off effect of the unappealing 'Metro' tiled interface. Rounded corners are slated for a future Windows 10 release.

At the time of writing, Huawei continues to advertise its high-end P30 Pro smartphone in the British press, adding that it's 'business as normal', despite the trade spat with the Trump administration. No one really knows for sure whether Google will be forced to withhold future support and updates for apps and the Android OS found on current or future Huawei phones.

Amazon's smartspeaker is to offer British users medical advice gleaned from the National Health Service website. Alexa's hands-free voice-recognition and search system may help the elderly, partially sighted or those with disabilities who are unable to use the phone or visit their GP. The NHS website is at: **www.nhs.uk** and contains a plethora of professional, non-commercial advice and information. See you next month for more *Net Work!*

The author can be reached at:
alan@epemag.net



Accuracy better than 100 parts per billion

Part 1 – by Tim Blythman

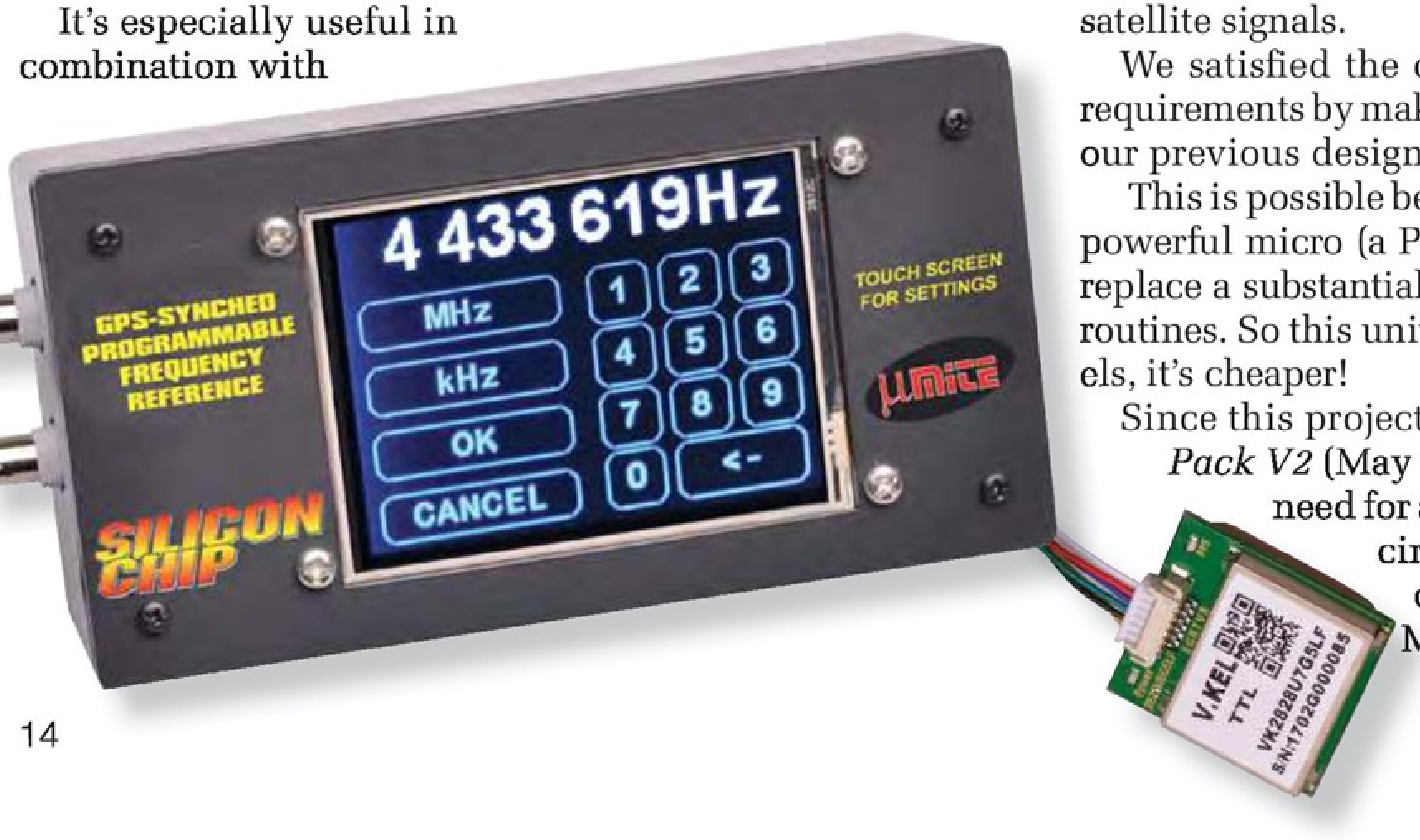
Lab Quality Programmable GPS-synced FREQUENCY REFERENCE

In design, service and repair, calibration or any other critical task in electronics, there are times when a really accurate and stable frequency reference is needed. The chances are, whatever frequency you require, this Micromite BackPack-based project can provide it. Exactly!

Our new GPS-based Frequency Reference provides three high accuracy, customisable outputs which are set up using a touchscreen and synchronised to the 1pps signal from a GPS module. Two of these can be set to a wide range of frequencies over the range of about 1-100MHz. The onboard temperature-compensated oscillator is within a temperature-controlled oven for maximum stability.

Not only is its GPS-synchronised oscillator very stable, but also you can choose from a huge range of possible output frequencies – in fact, it has three separate outputs, so you can even produce more than one signal at a time.

It's especially useful in combination with frequency counters, oscilloscopes and spectrum analysers with external reference inputs, especially if their inbuilt oscillator is not terribly accurate.



For example, many devices need a 10MHz reference and this unit can produce that exact frequency. But it's also quite common to see test equipment needing some oddball frequencies, so you can set up one of the outputs to cater to those too.

We've gone to quite some length to make this design not only very accurate and flexible, but also compact, easy to build, easy to use and relatively inexpensive. And we have also addressed most of the criticism levelled at our previous design, mainly to do with its ability to reject jitter from the satellite signals.

We satisfied the compactness and ease of construction requirements by making this unit considerably simpler than our previous design (April-May 2009).

This is possible because we are using a considerably more powerful micro (a PIC32) which means we've been able to replace a substantial portion of the hardware with software routines. So this unit is not only better than the earlier models, it's cheaper!

Since this project is based on the *Micromite LCD BackPack V2* (May 2018), the touchscreen eliminates the need for all the extra display and button-sensing circuitry and therefore the need for many cutouts in the front panel. Our usual Micromite supplier (**micromite.org**) can supply a kit to build the *BackPack V2* module; see the parts list for details.

The simplification of the circuitry has also meant that we can fit the extra circuitry on a much smaller board. In fact, it's now a similar size to the *BackPack V2* and simply stacks behind it, so the whole lot will easily fit into a small UB3 Jiffy box.

Plus, the touchscreen means we can make the user interface much easier to understand and simpler to use, with a lot of extra features.

But it's the triple PLL IC which provides the cherry on the cake – that is, the three configurable outputs. One of the outputs is limited to a small range of frequencies (including that all-important 10MHz option) but for the other two, you can type in just about any frequency and chances are, it will produce that exact frequency, or something very close. And if it isn't exact, it will tell you the difference.

Excellent long-term precision

The jitter reduction mentioned earlier is possible because we are no longer using a phase-locked-loop (PLL) IC to lock the voltage controlled oscillator (VCO) onto the GPS 1pps signals. A PLL will adjust the VCO frequency immediately upon detection of a phase difference, but that could just be due to jitter. Instead, we're using a software algorithm which averages the VCO frequency over a range of time periods and makes small, calculated adjustments to the frequency. We'll have more on that algorithm in the next article in this series.

Also, for it to be as effective as possible, we need an extremely stable oscillator. This has been achieved by putting a temperature-compensated oscillator in a temperature-controlled oven, along with a very stable reference regulator which is used to derive the VCO's control input voltage, along with a DAC that can produce very small voltage steps.

The end result is that with a satellite-derived 1pps signal with sufficient long-term accuracy, we can trim the VCO's output so that its error is measured not just in parts-per-million… but in parts-per-billion.

Features and specifications

- Main 40MHz oscillator is disciplined from GPS signals
- Accuracy of around ±100ppb after 30 minutes
- Three BNC outputs with programmable frequencies: ~ 1–100MHz
- Main oscillator is temperature compensated and oven regulated
- MicroMite BackPack-based 320 x 240 pixel LCD touchscreen for configuration and status
- Compact unit, 160 x 68 x 44mm overall (in UB3 Jiffy box)
- Powered from 5V USB supply @ 400mA (500mA at start-up)
- Optional disciplined 1pps output
- Works with a wide range of GPS units, with external or internal antenna
- Low parts count
- Modest parts cost
- Most parts are easy-to-solder SMDs.

Since the new unit is less complex, we've also managed to make it use much less power. Rather than needing a 12V supply, it runs off 5V instead. That simplifies the power supply and the unit simply runs from any USB (5V) power source such as a phone charger or even directly from a computer.

Principle of operation

The block diagram in Fig.1 shows the basic principle of the *Frequency Reference*. The oven section is shown at upper left, and inside it, there is a voltage-controlled oscillator (VCO1), a DS18B20 temperature sensor and transistor Q1. The Micromite controls the output voltage of the DAC (digital-to-analogue converter – IC6) using an SPI (serial peripheral interface) bus. This voltage affects the dissipation in transistor Q1 and it heats up the components inside the oven, including the digital temperature sensor.

The temperature reading is fed back to the Micromite over a Dallas 1-wire serial bus, and this information is used to adjust the voltage fed to Q1, regulating the oven temperature.

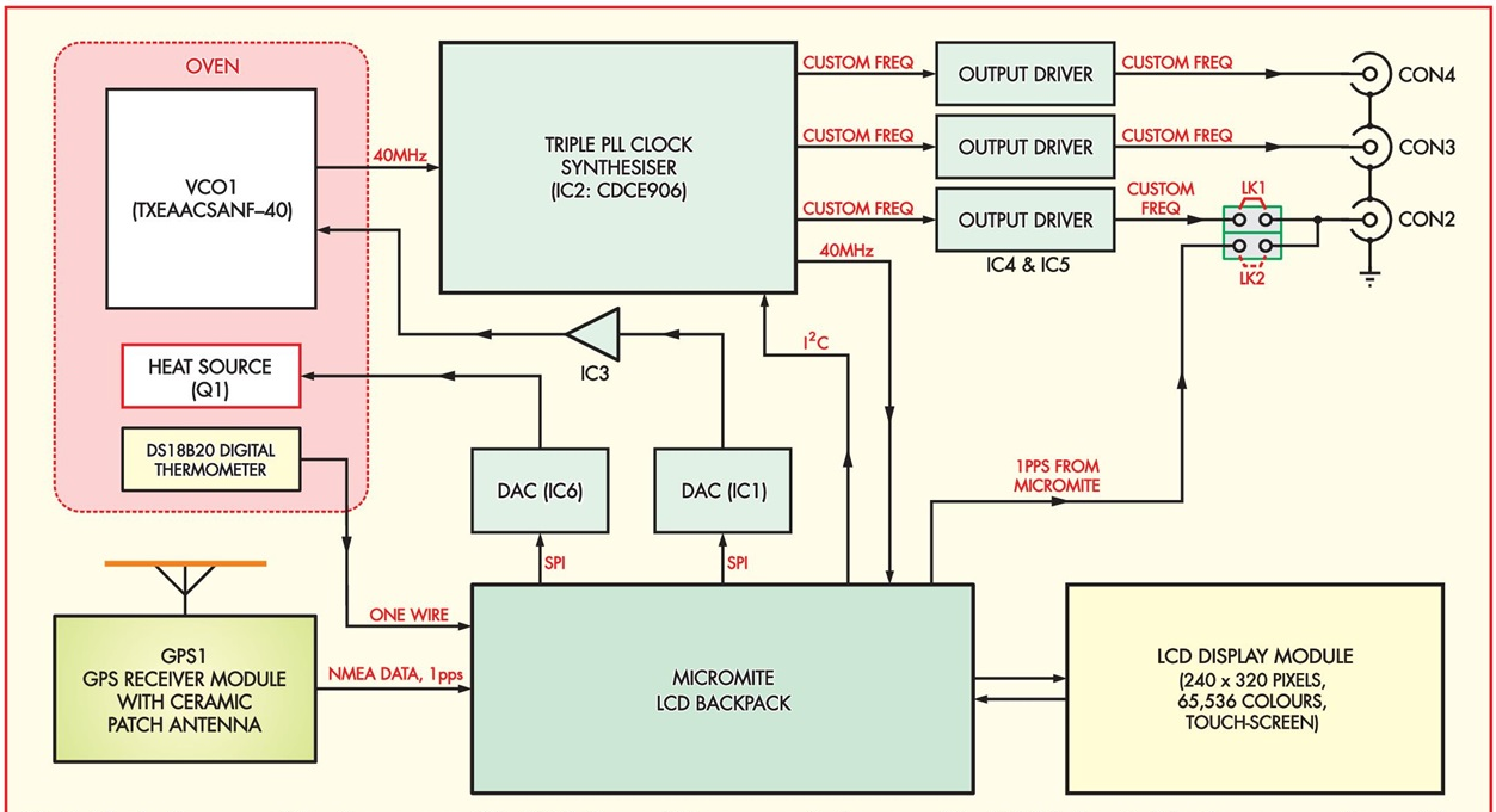
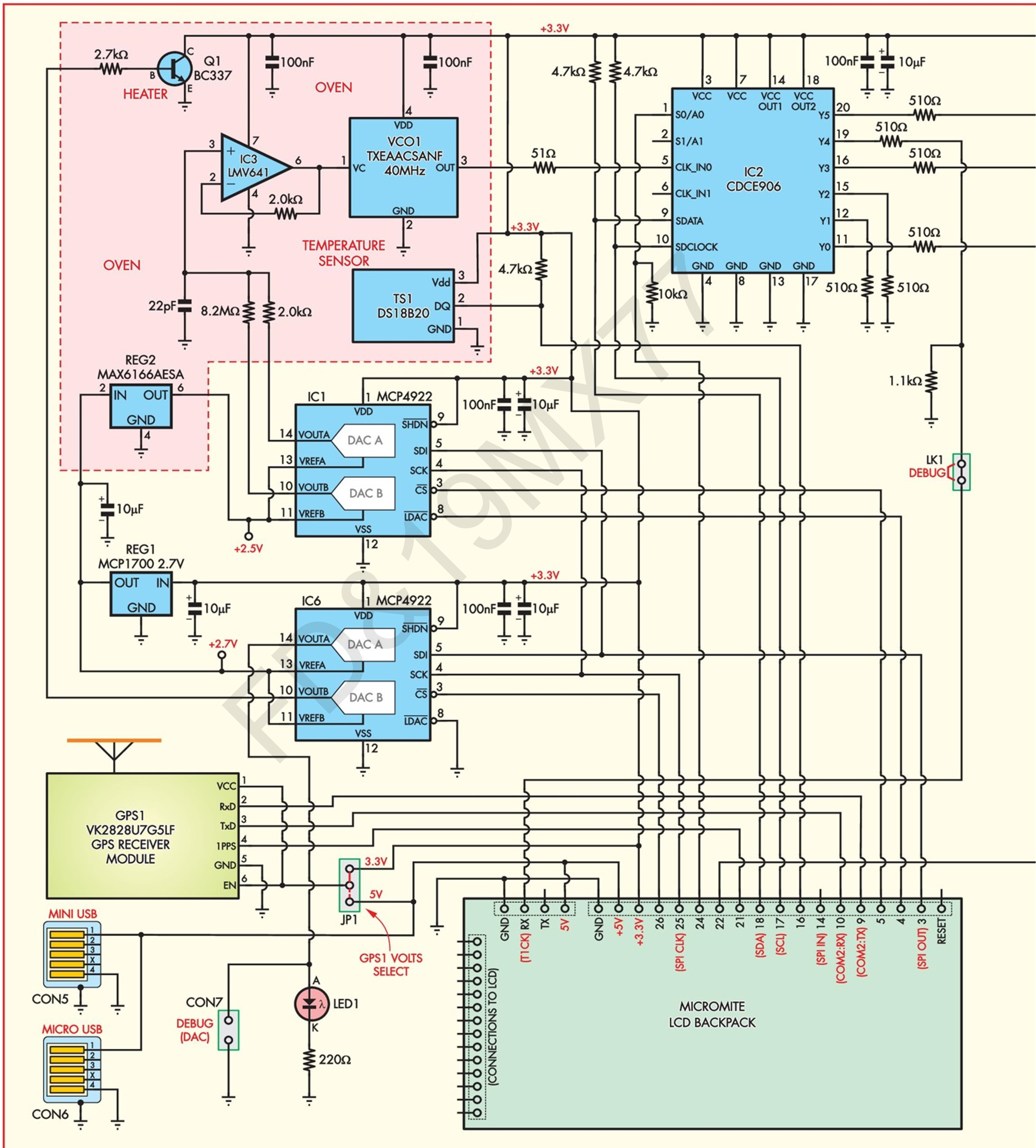

OVEN
VCO1
(TXEAACSANF–40)
HEAT SOURCE
(Q1)
DS18B20 DIGITAL
THERMOMETER
40MHz
TRIPLE PLL CLOCK
SYNTHESISER
(IC2: CDCE906)
CUSTOM FREQ
CUSTOM FREQ
CUSTOM FREQ
40MHz
OUTPUT DRIVER
OUTPUT DRIVER
OUTPUT DRIVER
IC4 & IC5
CUSTOM FREQ
CUSTOM FREQ
CUSTOM
FREQ
LK1
LK2
CON4
CON3
CON2
IC3
I²C
DAC (IC6)
DAC (IC1)
SPI
SPI
1PPS FROM
MICROMITE
ONE WIRE
NMEA DATA, 1pps
GPS1
GPS RECEIVER MODULE
WITH CERAMIC
PATCH ANTENNA
MICROMITE
LCD BACKPACK
LCD DISPLAY MODULE
(240 x 320 PIXELS,
65,536 COLOURS,
TOUCH-SCREEN)

Fig.1: block diagram of the *Programmable GPS-Synced Frequency Reference*. The VCO is located in the oven section, with transistor Q1 and a DS18B20 thermometer used to maintain it at a fixed temperature. This feeds the PLL, which then routes the signal to the Micromite and to the three outputs. The Micromite chip can then send commands to DAC IC1 to shift the VCO frequency via buffer IC3, to discipline the 40MHz VCO frequency using 1Hz pulses from the GPS module.

A second SPI DAC (IC1) produces a voltage which is buffered by IC3 and then fed to the VCO, to shift its output frequency slightly, over the range of 39,999,800-40,000,200Hz.

The initial tolerance of the VCO is ±2ppm which equates to ±80Hz, with a drift of up to ±1ppm (±40Hz) per year, and some small additional variation with temperature or supply voltage changes. By shifting the VCO frequency, we can compensate for these errors, getting its output very close to exactly 40,000,000Hz. Once the oven reaches the target temperature, it varies over a range of less than 1°C, improving the stability of the VCO into the parts-per-billion range.

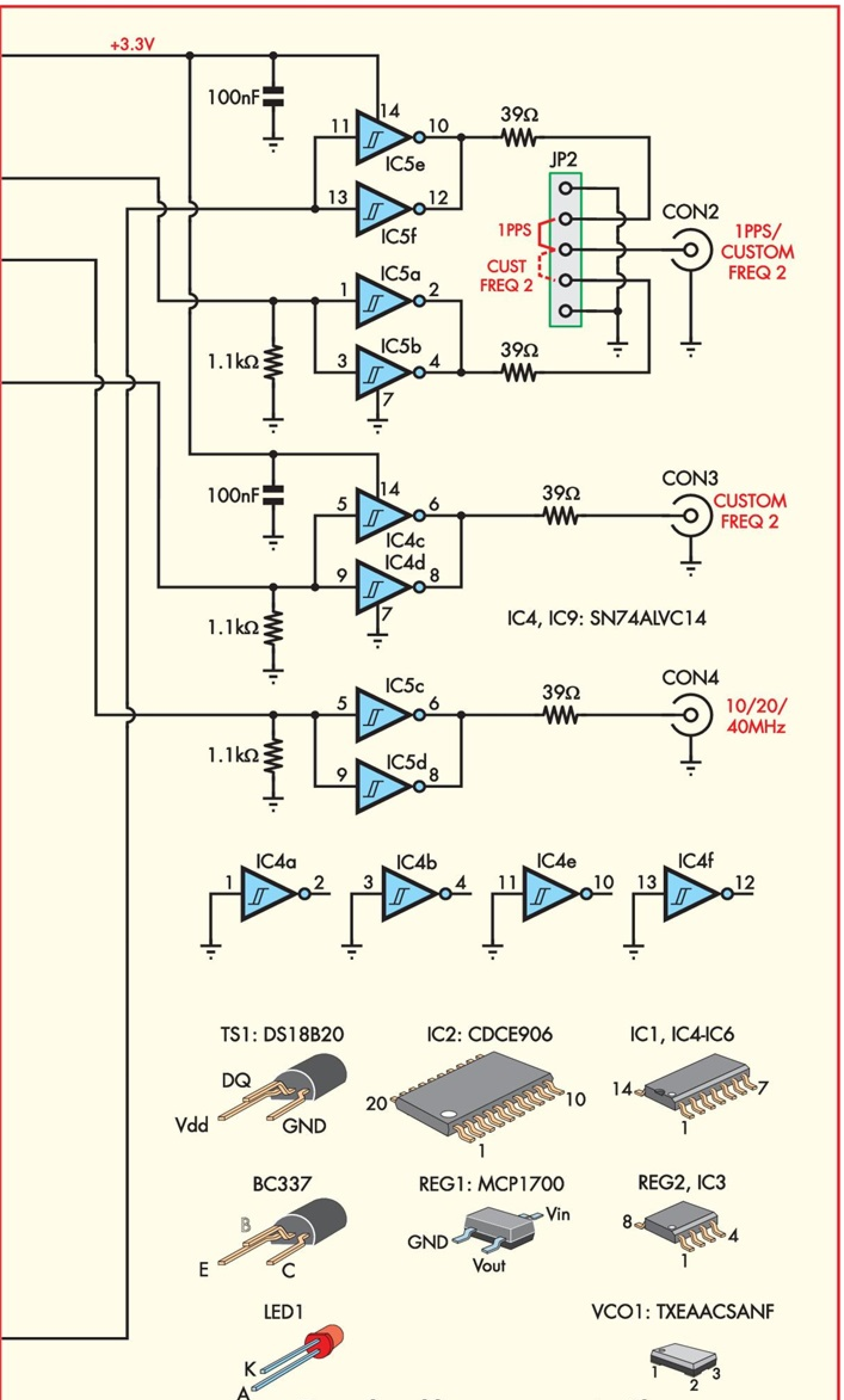
The trimmed frequency from the VCO is fed to a multi-PLL chip (IC2) and from there back to a high-frequency counter within the Micromite. The Micromite can then count the number of pulses received between each 1pps pulse from the GPS module to determine whether the VCO's


+3.3V
2.7kΩ
Q1
BC337
B
C
E
HEATER
100nF
OVEN
100nF
4.7kΩ
4.7kΩ
100nF
10µF
3
7
14
18
VCC
VCC
VCC OUT1
VCC OUT2
20
Y5
510Ω
1
S0/A0
2
S1/A1
19
Y4
510Ω
5
CLK_IN0
IC2
CDCE906
16
Y3
510Ω
6
CLK_IN1
15
Y2
9
SDATA
12
Y1
10
SDCLOCK
11
Y0
510Ω
GND
GND
GND
GND
4
8
13
17
10kΩ
510Ω
510Ω
3
IC3
LMV641
7
6
2
4
2.0kΩ
1
VCO1
TXEAACSANF
40MHz
VC
OUT
VDD
GND
4
2
3
51Ω
+3.3V
TEMPERATURE SENSOR
4.7kΩ
TS1
DS18B20
Vdd
DQ
GND
3
2
1
OVEN
22pF
8.2MΩ
2.0kΩ
REG2
MAX6166AESA
2
IN
OUT
6
GND
4
IC1
1 MCP4922
VDD
14
VOUTA
DAC A
13
VREFA
10
VOUTB
DAC B
11
VREFB
VSS
12
SHDN
9
SDI
5
SCK
4
CS
3
LDAC
8
100nF
10µF
+3.3V
1.1kΩ
LK1
DEBUG
10µF
+2.5V
REG1
MCP1700 2.7V
OUT
IN
GND
10µF
+2.7V
IC6
1 MCP4922
VDD
14
VOUTA
DAC A
13
VREFA
10
VOUTB
DAC B
11
VREFB
VSS
12
SHDN
9
SDI
5
SCK
4
CS
3
LDAC
8
100nF
10µF
+3.3V
GPS1
VK2828U7G5LF
GPS RECEIVER
MODULE
VCC 1
RxD 2
TxD 3
1PPS 4
GND 5
EN 6
3.3V
5V
JP1
GPS1 VOLTS SELECT
MINI USB
1
2
3
X
4
CON5
MICRO USB
1
2
3
X
4
CON6
CON7
DEBUG (DAC)
A
LED1
K
220Ω
GND
RX (T1CK)
TX
5V
GND
+5V
+3.3V
26
25 (SPI CLK)
24
22
21
18 (SDA)
17 (SCL)
16
14 (SPI IN)
10 (COM2:RX)
9 (COM2:TX)
5
4
3 (SPI OUT)
RESET
(CONNECTIONS TO LCD)
MICROMITE
LCD BACKPACK

FD619MX77

Programmable GPS-synced Frequency Reference frequency is spot on. If it is not, the Micromite adjusts the drive to the VCO to bring it back in line with the GPS pulses.

Note that it can also average its readings over a longer period to reduce the influence of jitter in the GPS 1pps pulses on the output of the VCO. The 40MHz frequency from the VCO is also fed to three programmable PLLs within IC2, which can be configured to produce a wide range of different frequencies. The outputs of the three PLLs are then buffered by IC4 and IC5, to provide a low impedance for BNC-socket output connectors CON2-CON4. If LK2 is fitted instead of LK1, CON2 will instead be fed with a disciplined (ie, more accurate) 1pps signal from the Micromite, instead of the third PLL output.



Reproduced by arrangement with
SILICON CHIP magazine 2019.
www.siliconchip.com.au

Fig.2: complete circuit diagram for the *Frequency Reference*, showing the *Micromite BackPack V2* as a 'black box' (the May 2018 article has the details). The 40MHz signal from VCO1 is fed to PLL IC2, which then produces the three programmable frequency output signals fed to CON2-CON4, as well as the signal fed back to RX/T1CK on the *BackPack*. This is divided down and compared to the 1pps signal from the GPS module at pin 21; and the difference is used to change the voltage at the outputs of dual DAC IC1, which are combined and buffered by IC3 and then fed to the VCO's control input.

Circuit description

Turning now to the full circuit diagram in Fig.2, you can see that there is a little more to it than described above.

Although oscillator VCO1 (TXEAACSANF-40) is in the oven, we also decided to use a temperature-compensated oscillator, for extra stability. IC1, the DAC controlling the VCO frequency, is an MCP4922 dual 12-bit type. We need fine control over the voltage fed to the VCO, so we can shift its frequency by very small amounts.

Rather than use an expensive 24-bit DAC, we are instead averaging the two outputs of the dual 12-bit DAC using different resistor values, so that the pin 10 output gives fine control and the pin 14 output adjusts the frequency in larger steps. With this arrangement, we can vary the frequency of the VCO over its full range, in steps of around 60ppb.

So that the VCO output is stable, the control voltage must also be very stable, so both the averaging resistors and the 2.5V reference for the DAC (REG2) are inside the oven, indicated by the shaded area on the circuit diagram.

The combined DAC output is fed to an LMV641 op-amp (IC3) so that the VCO's control input doesn't load up the averaging resistors and cause the voltage to shift, and also so that the VCO control input is driven from a low-impedance source to ensure it operates correctly. This is a high-precision, low-noise, low-power op amp. A 22pF capacitor at its input reduces unwanted noise.

While IC3 has no gain, there is a 2kΩ resistor in its feedback path so that the impedance seen by both inputs (pins 2 and 3) is the same. This is important, because mismatched input impedances cause increased thermal drift in op amps, and that is something we definitely do not want. DAC IC1's reference inputs (pin 11 and 13) are fed with 2.5V from REG2, a MAX6166A precision regulator with 5ppm/°C temperature coefficient.

Since its temperature variation is within 1°C and the VCO 'pulling range' is ±5ppm, that means the influence of variations in REG2's output should be no more than (2 × 5ppm) × 5ppm × 1°C, which is a fraction of a part per billion.

Also, REG2 is fed from a 2.7V regulated rail provided by REG1, itself a very stable 2.7V low-dropout (LDO) regulator, so its line regulation should be excellent.

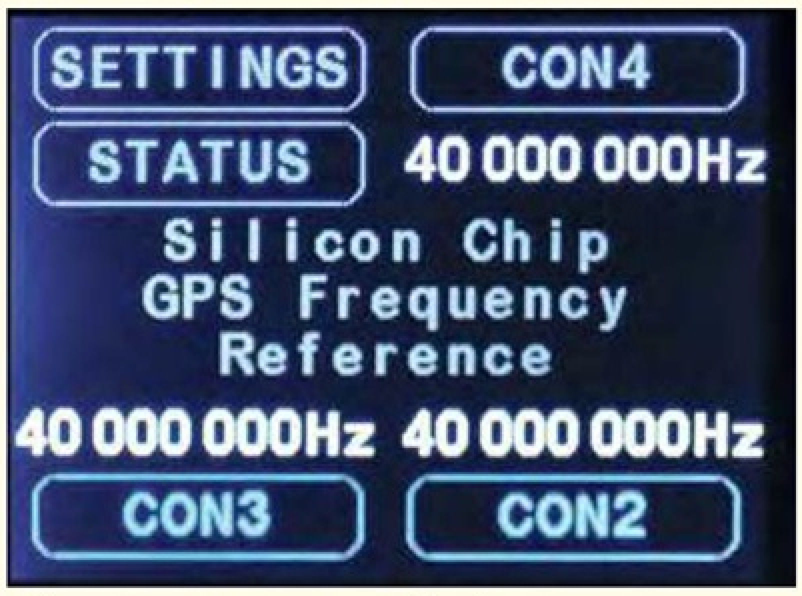
It should be apparent that we have gone to a lot of effort to ensure that the VCO's output is extremely stable and only shifts when the micro wants it to.

NPN transistor Q1, which provides heat for the oven, is connected directly across the 3.3V supply and so its dissipation is directly proportional to collector current, which in turn is related to its base current. The base current is determined by the voltage at output pin 10 of IC6, another MCP4922 dual SPI DAC, which uses a 2.7V reference from REG2 to avoid loading up REG1 unnecessarily.

Q1's 2.7kΩ base-current-limiting resistor was chosen to keep the maximum dissipation in Q1 to a safe level. The Micromite sets the DAC output to control the temperature, as measured by TS1, forming a closed control loop.

The second DAC in IC6 is not needed for any function related to the oven, so its output at pin 14 is connected to LED1 to vary its brightness as well as CON7, which is used for diagnostic purposes.

The DS18B20 oven temperature sensor (TS1) sends its data to the Micromite pin 16 and its output is fitted with a 4.7kΩ pull-up resistor, as required for the Dallas 1-wire protocol.

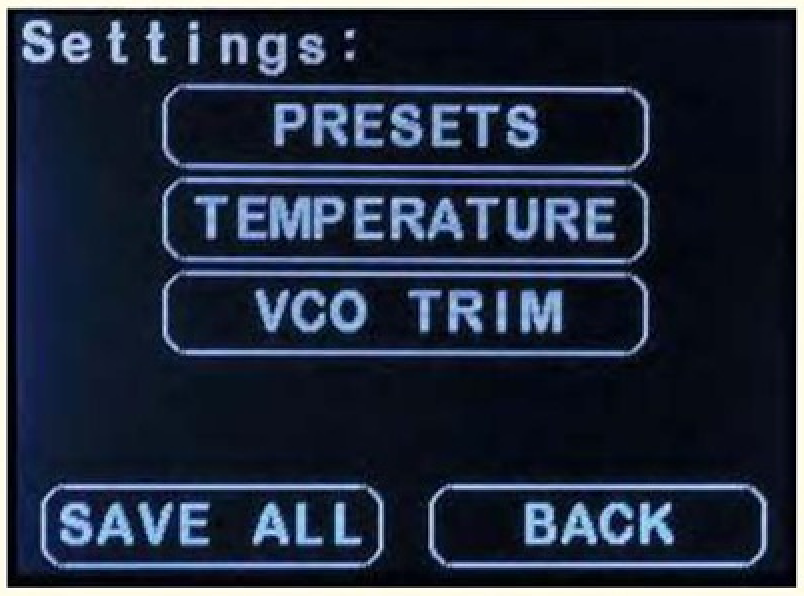


40 000 000Hz

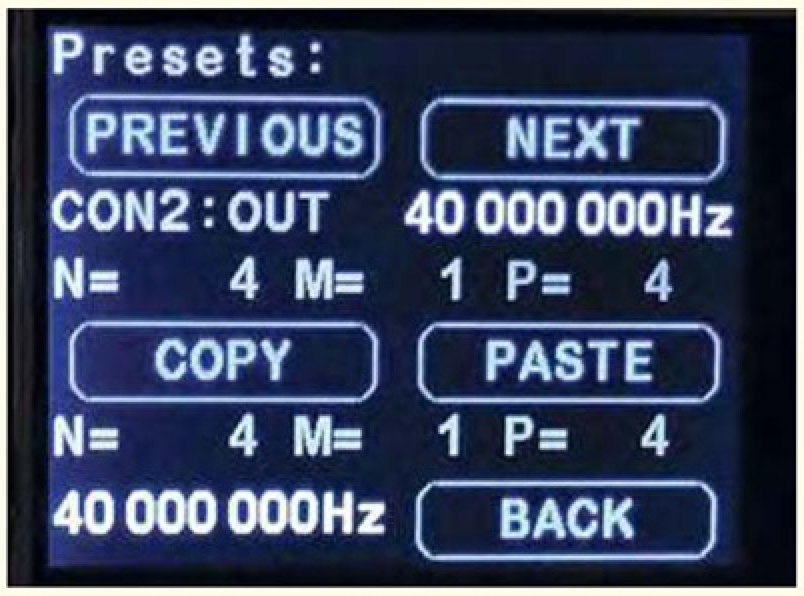
Silicon Chip
GPS Frequency
Reference

40 000 000Hz 40 000 000Hz

The Main Page provides buttons to access all the features of the *Frequency Reference*, as well as displaying the current state of the three outputs.



The settings are organised into three main groups: Presets, Temperature (for tuning the oven's operation) and VCO Trim (for manual adjustment of the VCO).



CON2:OUT 40 000 000Hz
N= 4 M= 1 P= 4

N= 4 M= 1 P= 4
40 000 000Hz

The Presets page allows frequency settings to be copied and pasted. All the presets are saved to non-volatile storage, so are preserved during power cycles.

Phase-locked loop IC

The PLL (IC2) is what allows us to have three customisable frequency outputs which are not simply integral fractions of the VCO's 40MHz frequency.

It is a Texas Instruments CDCE906 triple-PLL clock synthesiser IC, the only chip that is in a TSSOP (fine-pitch) SMD package because it is not available in a larger package.

It has six programmable outputs, but since it only has three internal PLLs, some settings are shared between them (we are using four of the six). It is controlled over an I^{2}C bus at pins 9 and 10, which are connected to pins 17 and 18 of the Micromite and 4.7kΩ pull-up resistors are fitted, as required by the I^{2}C standard.

If you are not familiar with PLLs, then (briefly) they consist of a voltage-controlled oscillator followed by a programmable divider.

A phase comparator compares the frequency and phase of the divided output to the input frequency and provides negative feedback, to adjust the oscillator frequency until the output of the divider matches the input, thus providing a fixed ratio between the input and output frequencies.

The 40MHz signal from VCO1 is fed into the three PLLs within IC2, via a 51Ω resistor, to provide IC2 with the expected 50Ω source impedance.

Each PLL has a multiplier and a divider, which allows a vast number of ratios to be chosen, and in turn, a wide range of frequencies to be derived from the input clock.

Each output can be programmed to take its input from any of the PLLs, and the PLL frequency can be further divided down to give an even wider range of output frequencies.

The two unused outputs of IC2, at pins 12 (Y1) and 15 (Y2) are terminated to ground with 510Ω resistors, while the other four outputs are fed to their destinations via 510Ω/1.1kΩ potential dividers.

Unfortunately, the CDCE906 data sheet was not specific about the output loading requirements and we found these resistor values necessary to provide reliable operation.

The dividers only reduce the output signal swing by about 33%, and since the chip runs off a 3.3V supply, that still gives a useful swing of around 2.2V peak-to-peak.

Output Y4 is fed back to the RX pin (pin 12) on the Micromite via jumper LK1, as this is the only pin which can measure frequencies this high (as described below).

Since this prevents the serial console from operating, LK1 has been provided; simply remove the jumper to access the serial console and re-insert it to allow normal operation.

The other three remaining outputs at pins 16, 19 and 20 (outputs Y3-Y5) are fed to paralleled pairs of gates in hex inverter chips IC4 and IC5. These are 74ALVC14 devices, which are a modern equivalent to the 74HC14 hex schmitt trigger inverter.

The difference is that these chips can operate at lower voltages (1.65-3.6V) and much higher frequencies (up to about 100MHz). They can also source or sink up to 50mA per output.

Thus, each pair can supply up to 100mA and they feed the outputs via 39Ω impedance-matching/current-limiting resistors.

A dedicated pair of inverters (IC5e/f) is also provided to buffer the disciplined 1pps output from pin 22 of the Micromite.

Four of the twelve inverter stages are unused (IC4a/b/e/f), so their inputs are tied to ground to prevent oscillation.

The link arrangement to select the signal source for CON2 is a little more complex than was shown on the block diagram. A 5-pin header is used, which allows a jumper shunt to be placed over either pins 2 and 3, or 3 and 4 to feed either signal through to the central pin of the BNC socket. But since pins 1 and 5 are wired to ground, this gives you the option of feeding either or both signals to off-board connectors by placing a 2-wire plug over pin pairs 1 and 2, and 4 and 5.

GPS module interface

The GPS module's serial TX and RX pins are wired directly to pins 9 and 10 on the Micromite so that the latter can receive NMEA serial data.

A jumper is placed on LK1 to select between a 3.3V or 5V power supply for the GPS module.

We used the trusty old VK2828U7-G5LF GPS module with onboard ceramic antenna, which can operate from either 3.3V or 5V.

Some constructors will prefer to use a module with an external antenna, and in this case, you may need to choose a particular voltage to suit the module used.

Whichever module you select, make sure it has a TTL serial interface along with a 1pps output pin.

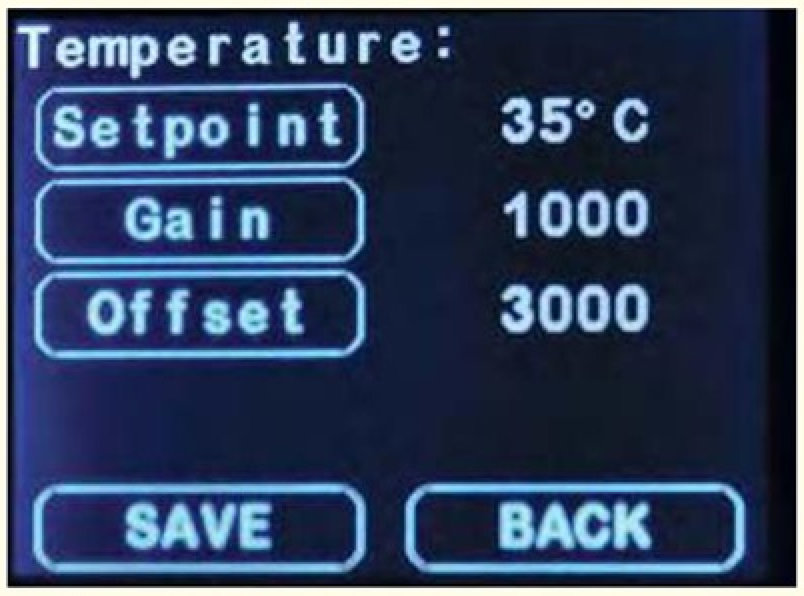
The NMEA output from the GPS module is a stream of serial text data which contains satellite information, including the time, date and location. The module also produces a 1pps pulse, which is fed directly to pin 21 on the Micromite.

Once the Micromite confirms from the NMEA data that a valid satellite fix has been achieved, it starts timing the 1pps pulses in order to discipline the VCO.

Power supply

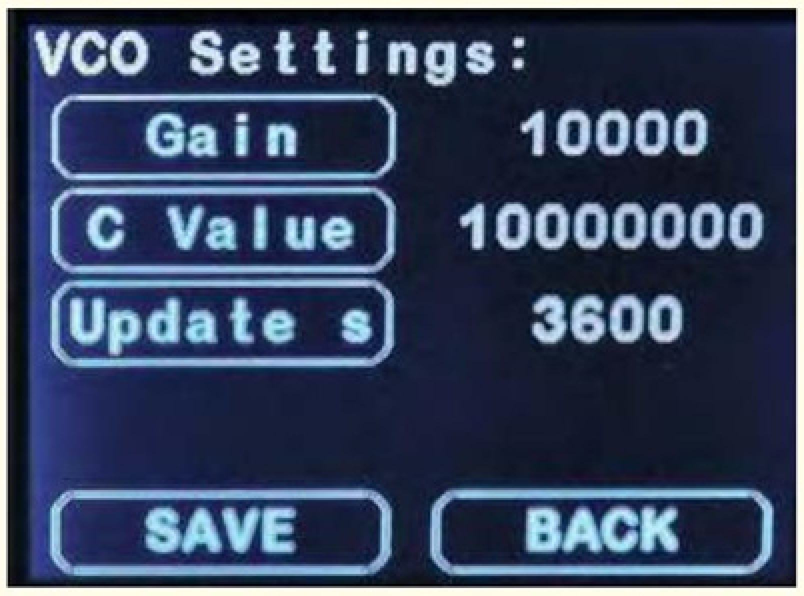
You can fit either a mini USB (CON5) or micro USB (CON6) socket and power then comes from a 5V USB charger or similar.

The 5V supply is fed directly to the *Micromite BackPack V2*, where it powers the LCD touchscreen and is regulated to 3.3V to power the Micromite chip. The 3.3V supply is then fed back to the *Frequency Reference* board, to power the remaining circuitry.



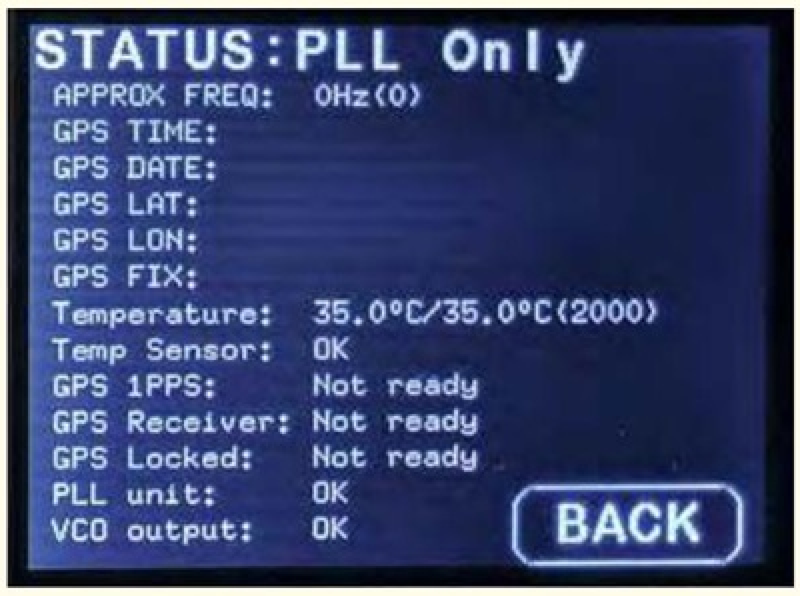
Temperature:
Setpoint 35° C
Gain 1000
Offset 3000

Tuning the temperature settings can be done on this page, although we found the initial values we tried were quite good. You may need to raise the setpoint if your workspace consistently gets above 35°C.



VCO Settings:
Gain 10000
C Value 10000000
Update s 3600

The VCO settings are basic, but very useful. Changing the C Value (control value) allows the VCO to be manually tuned, perfect if you have an atomic clock source for calibration.



```
STATUS:PLL Only
APPROX FREQ:   0Hz(0)
GPS TIME:
GPS DATE:
GPS LAT:
GPS LON:
GPS FIX:
Temperature:   35.0°C/35.0°C(2000)
Temp Sensor:   OK
GPS 1PPS:      Not ready
GPS Receiver:  Not ready
GPS Locked:    Not ready
PLL unit:      OK
VCO output:    OK
```

A Status page is provided to give information about how the *Frequency Reference* is performing. In this case, no GPS module is connected, and the unit has fallen back to PLL-only mode.

The only component on the *Frequency Reference* which may draw directly from the 5V supply is the GPS module, and that's only if you have selected the 5V option.

Everything else runs from 3.3V, with the 2.7V (REG1) and 2.5V (REG2) regulators only being used to provide the DAC reference voltages.

Remaining components

By default, pin 1 of IC2 (S0/A0/CLK_SEL) is configured to disable the six outputs when low. So we have fitted a 10kΩ resistor to ground to make the outputs disabled by default.

This pin is wired back to Micromite pin 24 so it can enable the PLL outputs by making that pin a digital output and setting it high.

All ICs have 100nF bypass capacitors between their main supply pins and ground, for reliable operation; and all regulators have 10µF ceramic input bypass and output filter capacitors, except for REG2, which has internal compensation, so does not require a capacitor on its output.

To minimise noise from DACs IC1 and IC6, and jitter from PLL IC2, these three devices also have 10µF ceramic bypass capacitors near their supply pins, in parallel with the 100nF capacitors.

Micromite BackPack

The Micromite is responsible for coordinating the functions of all the components on the GPS *Frequency Reference* board. As mentioned earlier, the two DACs IC1 and IC6 use the SPI bus, which is on pins 3 (SPI OUT) and 25 (SPI CLK) of the Micromite. This same bus is also used on the *Micromite BackPack V2* for communications with the touchscreen, but at different times, so the functions do not interfere.

The Micromite's digital output pin 5 is pulled low when the software sends an SPI command to IC1 and this is wired to its chip select ($\overline{CS}$) input. Similarly, Micromite pin 26 drives the $\overline{CS}$ input of DAC IC6.

Micromite pin 4 is wired to the $\overline{LDAC}$ input of IC1 (pin 8), which is driven to ensure that the output voltages of both DACs within IC1 change at the same time. This prevents glitches from changes in the DAC output voltage being propagated to the input of VCO1.

Communications with the DACs is simple since the micro only needs to send the new digital value over the SPI bus and the output voltages then change in response.

The PLL IC has a more complex interface and is controlled by programming an array of internal registers which have various functions.

Note that we had to use the Micromite's pin 26 beause there are no free pins on the I/O header (pin 14 is a shared SPI pin, so cannot be used). The *Micromite BackPack V2* uses pin 26 for the optional software LCD backlight control, so *Micromite BackPack V2* must be built with hardware (trimpot) backlight brightness control instead.

Parts list – Precision Frequency Reference

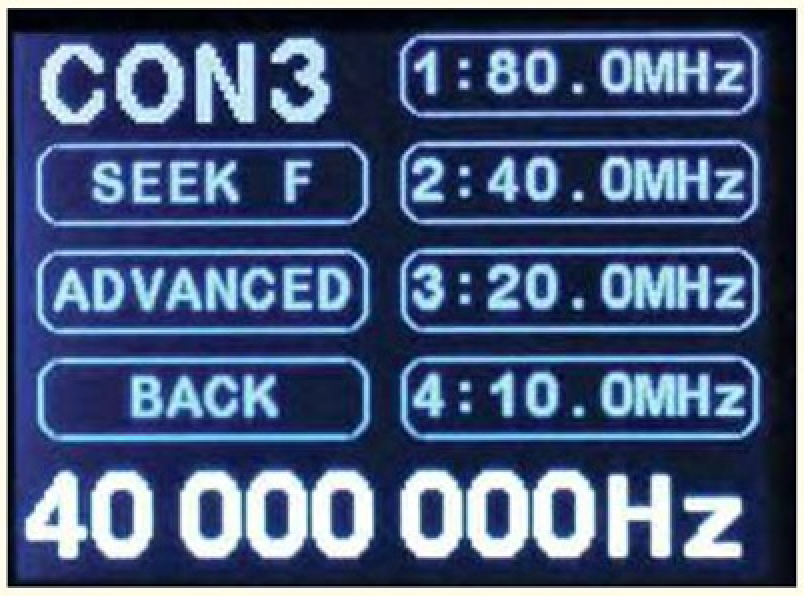
1 double-sided PCB, code 04107181, 120mm x 55mm
1 Micromite BackPack V2 kit (described in May 2018) [available from micromite.org]
1 VK2828U7G5LF or equivalent GPS module [Banggood ID: ID: 965540]
3 PCB-mount BNC sockets (CON2-CON4) [Jaycar PS0661]
1 UB3 Jiffy Box
1 40-way snappable pin header (GPS1,JP1,JP2,LK1) [Altronics P5430, Jaycar HM3212]
3 jumper shunts (shorting blocks) (JP1,JP2,LK1)
1 SMD USB mini-B (CON5) or micro-B (CON6) socket
3 M3 tapped 12mm-long nylon spacers
3 M3 x 20mm pan-head machine screws
1 USB charger or similar power supply with cable to suit CON5/CON6

Parts for oven enclosure
2 100mm cable ties
2 bottle caps, film canisters, small foam cups or similar

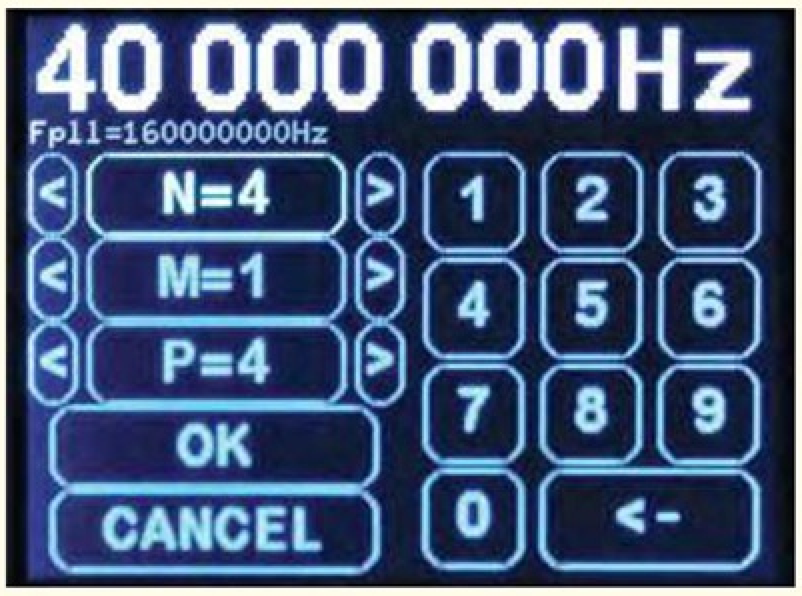
Semiconductors
2 MCP4922 dual 12-bit SPI DAC ICs, SOIC-14 (IC1,IC6)
1 CDCE906 programmable PLL/clock synthesiser, TSSOP-20 (IC2)
1 LMV641 low-power op amp, SOIC-8 (IC3)
2 SN74ALVC14 hex schmitt trigger inverter, SOIC-14 (IC4,IC5)
1 DS18B20 temperature sensor IC, TO-92 (TS1)
1 MCP1700-2.7V voltage regulator, TO-92 or SOT-23 (REG1)
1 MAX6166AESA 2.5V voltage regulator, SOIC-8 (REG2)
1 3mm LED (LED1)
1 BC337 NPN transistor, TO-92 (Q1)
1 TXEAACSANF-40 VCTCXO IC, 4-SMD (VCO1)

Capacitors (all SMD X7R, 3216/1206 size)
5 10µF 7 100nF 1 22pF

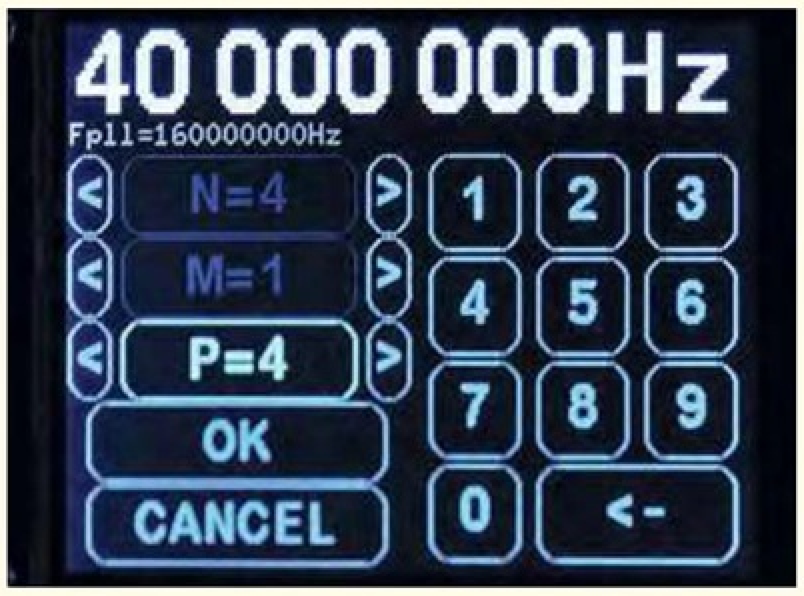
Resistors (all SMD 1%, 3216/1206 size)
1 8.2MΩ 1 10kΩ 3 4.7kΩ 1 2.7kΩ 2 2kΩ
4 1.1kΩ 6 510Ω 1 220Ω 1 51Ω 4 39Ω



Each output (CON2, CON3 and CON4) has its own control screen and allows setting of the output frequency by automatic seeking, manual adjustment of PLL parameters and from stored presets.



The advanced screen gives direct control of the N, M and P dividers, which are used by each PLL, as well as reporting whether the resulting PLL frequency is within the correct range.



CON4's advanced PLL control is restricted to modifying the P value, as N and M are locked to provide the 40MHz reference that is fed back into the Micromite.

Also, the Micromite must use its internal TIMER1 feature to keep track of the frequency, as none of the other timers are fast enough to manage the 40MHz signal. Unfortunately, the TIMER1 pin is attached in hardware to the Micromite's console RX on pin 12, so we have to shut down IC2 until it is needed.

Using the console RX pin to capture the 40MHz signal from the VCO means that we cannot use the console during normal operation, as this will cause the 40MHz signal to be swamped by signals from the console.

To get around this, the two USB sockets on the board (CON5 and CON6) are used for power only. Link LK1 is provided so that you can still program the Micromite via the serial console, and it is then shorted with a jumper for normal operation.

Controlling the PLL

IC2's software registers allow us to provide a comprehensive range of output frequencies – we can set up internal multipliers and dividers to determine a wide range of internal PLL frequencies. We have the capability to read and write these internal registers; the read function is used only to verify that the writes have occurred correctly.

Each PLL inside IC2 has a 12-bit 'N' divider and a 9-bit 'M' divider. But since the 'N' divider is in the feedback loop of the PLL, it actually has the effect of multiplying the frequency. It is the N/M fraction which determines the ratio between the PLL and input frequencies. Each of the six outputs also has a separate 7-bit 'P' divider. The N, M and P values are all integers (ie, whole numbers).

In more detail, the incoming (nominally 40MHz) frequency is multiplied by N and then divided by M to give the PLL frequency, and the PLL frequency is divided by P to arrive at the output frequency. The PLL frequency must be in the range of 80-300MHz, and the output frequency is limited to a range from 1-167MHz. Our output buffers limit the maximum usable frequency to around 100MHz.

We configure the registers in IC2 so that PLL1 feeds into CON2, PLL3 feeds into CON3 and PLL2 feeds into both CON4 and back to the Micromite's pin 12 frequency counter input. While PLL1 and PLL3 can be set to a wide range of frequencies, PLL2 is fixed to run at 160MHz, so that a 'P' divider of four gives us our 40MHz signal to feedback to the Micromite.

That means, however, that CON4 cannot be set to produce just any frequency. But it can still be set to a number of different frequencies, ie, 160MHz divided by an integer between two and 127 (80MHz, 53.3MHz, 40MHz, 32MHz, 26.7MHz, …, 1.26MHz).

While CON2 and CON3 can provide a much more flexible range of frequencies, the software actually has to do quite a bit of work to calculate the N, M and P values required to produce a specific frequency. After all, there are 268 million possible combinations [$2^{(12+9+7)}$].

While it's possible that there is no combination of values which will give a particular frequency that you want, chances are, if it is in the range of 10-100MHz then the unit will be able to produce something very close (and you will be able to see on the screen what that frequency actually is).

Since there are so many combinations of values, we had to carefully design an algorithm to find the best combination. It starts by determining what values of P are valid given the desired frequency. Because the PLL frequency must fall between 80MHz and 300MHz, this gives a fairly small range.

For example, to produce a 40MHz frequency, P must be between two (80MHz PLL) and seven (280MHz PLL). Since the value of P is limited to seven bits, the highest valid value for P is 127. This is also what determines the lowest possible frequency that the PLL output can produce, which is 630kHz (80MHz÷127).

Since the N and M registers have 12 and nine bits respectively, that puts an upper limit on their values at 4095 and 511 respectively. There is also a restriction that N must be greater than or equal to M, but given that the minimum PLL frequency is 80MHz, N must be greater than M to achieve this from a 40MHz source anyway.

Since M has a smaller range of values, the algorithm iterates over the valid values of P and M, works out what the exact (decimal) value of N would need to be to produce the desired frequency, and then tests the next highest and lowest integer values to see how close they would be to our target frequency.

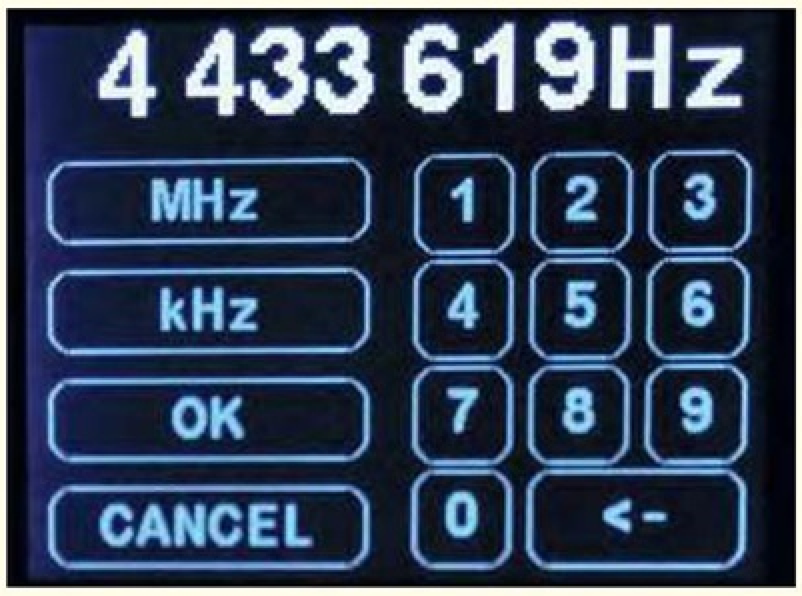
As the iteration occurs, if a better match is found, it is stored. If an exact match is found, then no further searching need be done. Otherwise, it continues until all viable PLL setting combinations have been tested.

Unfortunately, when this algorithm was implemented in BASIC, it took over a minute to complete, which is far too long.

Thus, we had to write a CFUNCTION to speed up the process. Once the C code has been compiled, it generates PIC32 machine code which is inserted into and can be called from the BASIC program, and this runs much faster. Our CFUNCTION version of the code takes less than one second to complete.

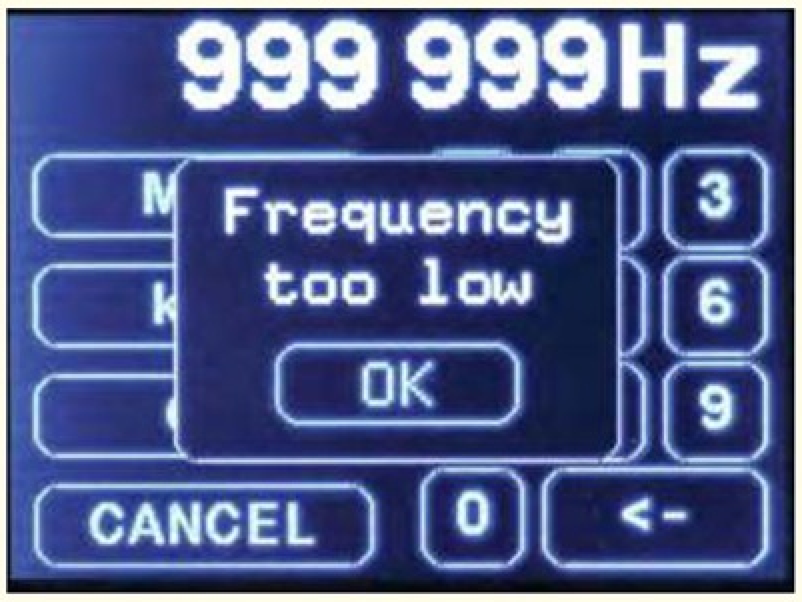
As well as the N, M and P values, there are a number of registers which need to be set up for the PLL to operate as required. Most of these are initially set by 'dumping' an array of data into the registers during the initialisation phase. After this, the N, M and P registers are about all that is changed by the program in the Micromite.

Each PLL also has a one-bit flag which can be used to select either a



4 433 619Hz

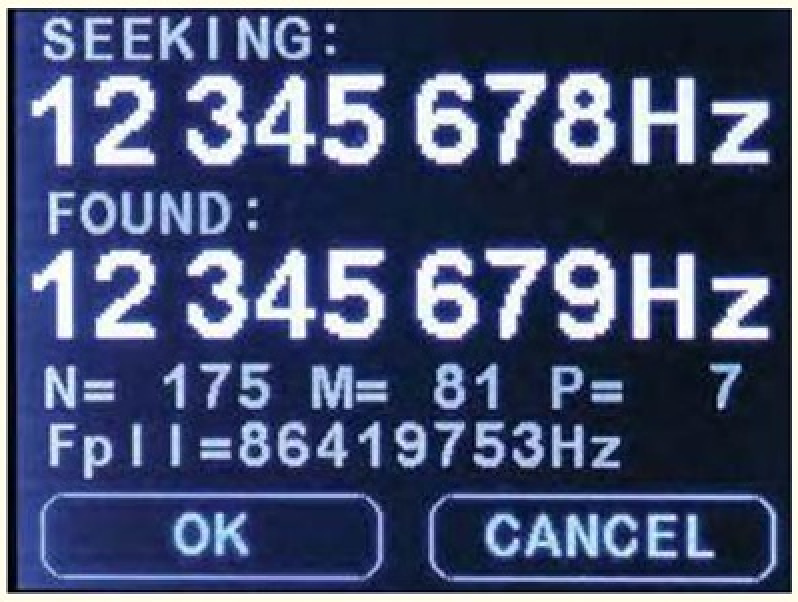
The Seek page allows the frequency to be entered by a numeric keypad and provides MHz and kHz shortcuts to speed up entry of custom frequencies.



999 999Hz

Frequency too low

If an entered frequency is too low for the Seek algorithm, a message box advises the fact and returns to the previous screen, preventing invalid results.



SEEKING:
12 345 678Hz
FOUND:
12 345 679Hz
N= 175 M= 81 P= 7
Fpll=86419753Hz

When a valid Seek frequency is entered, the Micromite finds the nearest frequency which can be generated by the PLL and displays it so that it can be checked before being output.

high-speed mode (above 180MHz) or low-speed mode (below 200MHz). As each group of registers are updated, this is set to the appropriate value.

As well as being able to enter a desired frequency and having the register values calculated for you, we decided to give users the option to enter the values of N, M and P manually. We found during testing that the restrictions stated in the data sheet on the PLL frequency are not hard-and-fast rules, and that the chip is able to operate at frequencies outside the specified range.

So our software does not enforce these rules for manually-entered values, although it does give a warning for combinations that would result in a PLL frequency outside the normal range.

Other software considerations

The rest of the software is relatively straightforward. Updating the DAC outputs only requires selecting the chip, writing 16 bits, then de-selecting the chip, which is just a few lines of code.

Processing and parsing the NMEA data from the GPS module is a bit more involved, as we have to check that the GPS module has a proper fix before trying to discipline the VCO from the 1pps signal (otherwise the 1pps signal may not be accurate). This involves checking that we have received the $GPRMC sentence and that it has the value 'A' at a certain point, and not 'V'.

Since it is not much more effort, we also decipher the GPS latitude, longitude, UTC time and date, and display these values on the status page as confirmation that the GPS module is working correctly.

Apart from the software algorithm for setting frequencies, we also had to create a second CFUNCTION to count the incoming 40MHz pulses from the VCO. This requires setting up the TIMER1 interrupt and an interrupt service routine to keep track of when the 16-bit hardware counter overflows (the 32-bit timers are not able to operate at this high a frequency).

The same routine also provides the disciplined 1pps output by toggling pin 22 every 20 million received pulses. So, the accuracy of the output 1pps signal is matched to the accuracy of the 40MHz oscillator, as they work in lockstep.

Developmental trials and tribulations

This project took some time to finish, and like many of our more ambitious projects we encountered a few stumbling blocks along the way. The final design is pretty close to the initial concept, but we had to make a few refinements for it to work properly.

As is typical, we are using all of the Micromite BackPack's free I/O pins. The only way to get more I/Os would have been to use a board with an SMD micro, but they are trickier to solder. In the end, the only real compromise we had to make was to use one of the console pins for I/O, making our debugging more difficult.

This was necessary because the PIC32 only provides one input pin for each timer clock and only one of the timers is fast enough to measure the 40MHz signal from the VCO. That pin just happens to coincide with the serial console transmit function.

So we had to use the touchscreen to display debugging messages. That meant we couldn't see BASIC error messages. While the V2 BackPack has a USB console connector, unlike the one on the Plus BackPack, this shares the same TX and RX pins, so that doesn't help.

The other major problems we had were with the PLL/Clock Synthesiser chip, IC2. It would randomly freeze up the I^{2}C bus, sometimes locking the Micromite and often just failing to respond at all. The I^{2}C signals from the Micromite looked fine on a scope.

After many hours of probing, it became apparent that the I^{2}C signals were fine, but the chip couldn't decode them because its ground was so noisy. We then noticed that the ground trace to the PLL/Clock Synthesiser chip had disappeared. This chip is closest to the Micromite header on the PCB and we had laid a short ground trace between them.

We realised that Altium Designer thinks it's smarter than us; it saw that there were two different ground paths between those pins and decided to remove one of them as it considered it redundant. So the chip's ground connection was via a long, circuitous path, hence the noise.

We also found that IC2 operated much more reliably when the outputs were loaded up with around 1.5kΩ to ground and with a 50Ω series resistor to set the source impedance for the clock input; this was not mentioned in the CDCE906 data sheet, so we had to figure it out ourselves.

Finally, we managed to simplify the design during development. We originally had two extra logic ICs, a divider to produce 20MHz and 10MHz signals from the 40MHz VCO output, and another to select which of these three signals was fed to CON4.

We later realised that all these functions could be done inside IC2, which would also allow for more frequency options for CON4. Since we had to revise the board to get IC2 to work properly, we eliminated the two extra ICs at the same time.

We came up with an easy way to do this accurately. Rather than letting the 16-bit timer roll over at 65,536 as it would normally (ie, after 2^{16} pulses), we set it to roll over after 62,500 pulses. Then, each time it rolls over, we increment another counter and once it reaches 320, that means that 20 million pulses (62,500 x 320) have occurred.

So we only need to determine whether to toggle the state of pin 22 right at the start of the interrupt handler routine, and since the delay will be the same each time, the duty cycle will be exactly 50% and the frequency will be exactly locked to the VCO.

The TIMER1 interrupt handler also increments a 64-bit counter by 62,500 each time it is called. Then, when a 1pps pulse from the GPS module is detected on pin 21, the TIMER1 value is added to this counter and that forms the timestamp, which is stored in a circular buffer.

The intervals between these time stamps are then fed into an algorithm to determine whether to adjust the VCO frequency, and if so, in which direction and by how much, to keep it running at exactly 40MHz, or as close as is possible.

User interface

As is usually the case with projects using a complex touch interface, the code to display information and process user input is quite involved and, including font data, takes up about half of the BASIC source code.

There are nine distinct interface screens, each quite different. The main overview page has five buttons, one to access the 'Settings' page, one for the 'Status' page and one each to set up the three outputs. The current frequencies at CON2, CON3 and CON4 are also displayed on this screen.

The output settings screen allows one of four preset frequencies to be loaded, or custom frequencies to be programmed, either using the automatic search algorithm or by manually setting the N, M and P divider values. A long press on one of the preset buttons allows the current output frequency value to be stored in a preset, while a short press loads that frequency immediately.

The settings page allows the presets to be copied between the various outputs. The characteristics and response of the temperature controller and VCO adjustment algorithm can also be changed.

The VCO's control voltage can be changed manually, so you can directly adjust the VCO frequency if you have access to a high-precision frequency reference for calibration. The adjustment interval can be changed too. Longer adjustment periods mean more data is available to perform a more accurate adjustment, but it will take longer to settle.

Physical construction

Like many of our Micromite projects that use the *Micromite LCD BackPack V2* board, the *Programmable GPS Frequency Reference* is designed to fit into a UB3 jiffy box, making for a very compact piece of test equipment.The BNC connectors project out the side of the box, with the front panel dedicated to the touchscreen.

Building this project is not particularly difficult, although there are a number of SMD parts. This is because IC2 is only available in an SMD (TSSOP) package, and we would have had to use a significantly larger box if we had used mostly through-hole parts. Except in the case of IC2, where we had no choice, we have selected mostly easy-to-solder (larger) SMDs.

The oven is made from just a few commonly available parts and does not take long. A few holes need to be drilled and cut into the plastic case, but once you have built the PCBs, the rest is pretty easy.

We'll get into the construction and operating details in *Part 2*, next month.

STEWART OF READING
17A King Street, Mortimer, near Reading, RG7 3RS
Telephone: **0118 933 1111** Fax: **0118 933 2375**
USED ELECTRONIC TEST EQUIPMENT
Check website **www.stewart-of-reading.co.uk**
(ALL PRICES PLUS CARRIAGE & VAT)
Please check availability before ordering or calling in

Fluke/Philips PM3092 Oscilloscope
2+2 Channel 200MHz Delay TB,
Autoset etc – £250

HP 54600B Oscilloscope
Analogue/Digital Dual Trace 100MHz
Only £75, with accessories £125

LAMBDA GENESYS	PSU GEN100-15 100V 15A Boxed As New	£400
LAMBDA GENESYS	PSU GEN50-30 50V 30A	£400
IFR 2025	Signal Generator 9kHz – 2.51GHz Opt 04/11	£900
IFR 2948B	Communication Service Monitor Opts 03/25 Avionics	POA
IFR 6843	Microwave Systems Analyser 10MHz – 20GHz	POA
R&S APN62	Syn Function Generator 1Hz – 260kHz	£295
Agilent 8712ET	RF Network Analyser 300kHz – 1300MHz	POA
HP8903A/B	Audio Analyser	£750 – £950
HP8757D	Scaler Network Analyser	POA
HP3325A	Synthesised Function Generator	£195
HP3561A	Dynamic Signal Analyser	£650
HP6032A	PSU 0-60V 0-50A 1000W	£750
HP6622A	PSU 0-20V 4A Twice or 0-50V 2A Twice	£350
HP6624A	PSU 4 Outputs	£400
HP6632B	PSU 0-20V 0-5A	£195
HP6644A	PSU 0-60V 3.5A	£400
HP6654A	PSU 0-60V 0-9A	£500
HP8341A	Synthesised Sweep Generator 10MHz – 20GHz	£2,000
HP83630A	Synthesised Sweeper 10MHz – 26.5 GHz	POA
HP83624A	Synthesised Sweeper 2 – 20GHz	POA
HP8484A	Power Sensor 0.01-18GHz 3nW-10µW	£75
HP8560E	Spectrum Analyser Synthesised 30Hz – 2.9GHz	£1,750
HP8563A	Spectrum Analyser Synthesised 9kHz – 22GHz	£2,250
HP8566B	Spectrum Analsyer 100Hz – 22GHz	£1,200
HP8662A	RF Generator 10kHz – 1280MHz	£750
Marconi 2022E	Synthesised AM/FM Signal Generator 10kHz – 1.01GHz	£325
Marconi 2024	Synthesised Signal Generator 9kHz – 2.4GHz	£800
Marconi 2030	Synthesised Signal Generator 10kHz – 1.35GHz	£750
Marconi 2023A	Signal Generator 9kHz – 1.2GHz	£700

Marconi 2305	Modulation Meter	£250
Marconi 2440	Counter 20GHz	£295
Marconi 2945/A/B	Communications Test Set Various Options	POA
Marconi 2955	Radio Communications Test Set	£595
Marconi 2955A	Radio Communications Test Set	£725
Marconi 2955B	Radio Communications Test Set	£800
Marconi 6200	Microwave Test Set	£1,500
Marconi 6200A	Microwave Test Set 10MHz – 20GHz	£1,950
Marconi 6200B	Microwave Test Set	£2,300
Marconi 6960B	Power Meter with 6910 sensor	£295
Tektronix TDS3052B	Oscilloscope 500MHz 2.5GS/s	£1,250
Tektronix TDS3032	Oscilloscope 300MHz 2.5GS/s	£995
Tektronix TDS3012	Oscilloscope 2 Channel 100MHz 1.25GS/s	£450
Tektronix 2430A	Oscilloscope Dual Trace 150MHz 100MS/s	£350
Tektronix 2465B	Oscilloscope 4 Channel 400MHz	£600
Farnell AP60/50	PSU 0-60V 0-50A 1kW Switch Mode	£300
Farnell XA35/2T	PSU 0-35V 0-2A Twice Digital	£75
Farnell AP100-90	Power Supply 100V 90A	£900
Farnell LF1	Sine/Sq Oscillator 10Hz – 1MHz	£45
Racal 1991	Counter/Timer 160MHz 9 Digit	£150
Racal 2101	Counter 20GHz LED	£295
Racal 9300	True RMS Millivoltmeter 5Hz – 20MHz etc	£45
Racal 9300B	As 9300	£75
Solartron 7150/PLUS	6½ Digit DMM True RMS IEEE	£65/£75
Solatron 1253	Gain Phase Analyser 1mHz – 20kHz	£600
Solartron SI 1255	HF Frequency Response Analyser	POA
Tasakago TM035-2	PSU 0-35V 0-2A 2 Meters	£30
Thurlby PL320QMD	PSU 0-30V 0-2A Twice	£160 – £200
Thurlby TG210	Function Generator 0.002-2MHz TTL etc Kenwood Badged	£65

HP33120A	Function Generator 100 microHz – 15MHz	£350
HP53131A	Universal Counter 3GHz Boxed unused	£600
HP53131A	Universal Counter 225MHz	£350
Audio Precision	SYS2712 Audio Analyser – in original box	POA
Datron 4708	Autocal Multifunction Standard	POA
Druck DPI 515	Pressure Calibrator/Controller	£400
Datron 1081	Autocal Standards Multimeter	POA
ENI 325LA	RF Power Amplifier 250kHz – 150MHz 25W 50dB	POA
Keithley 228	Voltage/Current Source	POA
Time 9818	DC Current & Voltage Calibrator	POA

HP/Agilent HP 34401A Digital
Multimeter 6½ Digit £325 – £375

Marconi 2955B Radio
Communications Test Set – £800

Win a Microchip PIC-IoT WG Development Board

Practical Electronics is offering its readers the chance to win a Microchip PIC-IoT WG Development Board (AC164164).

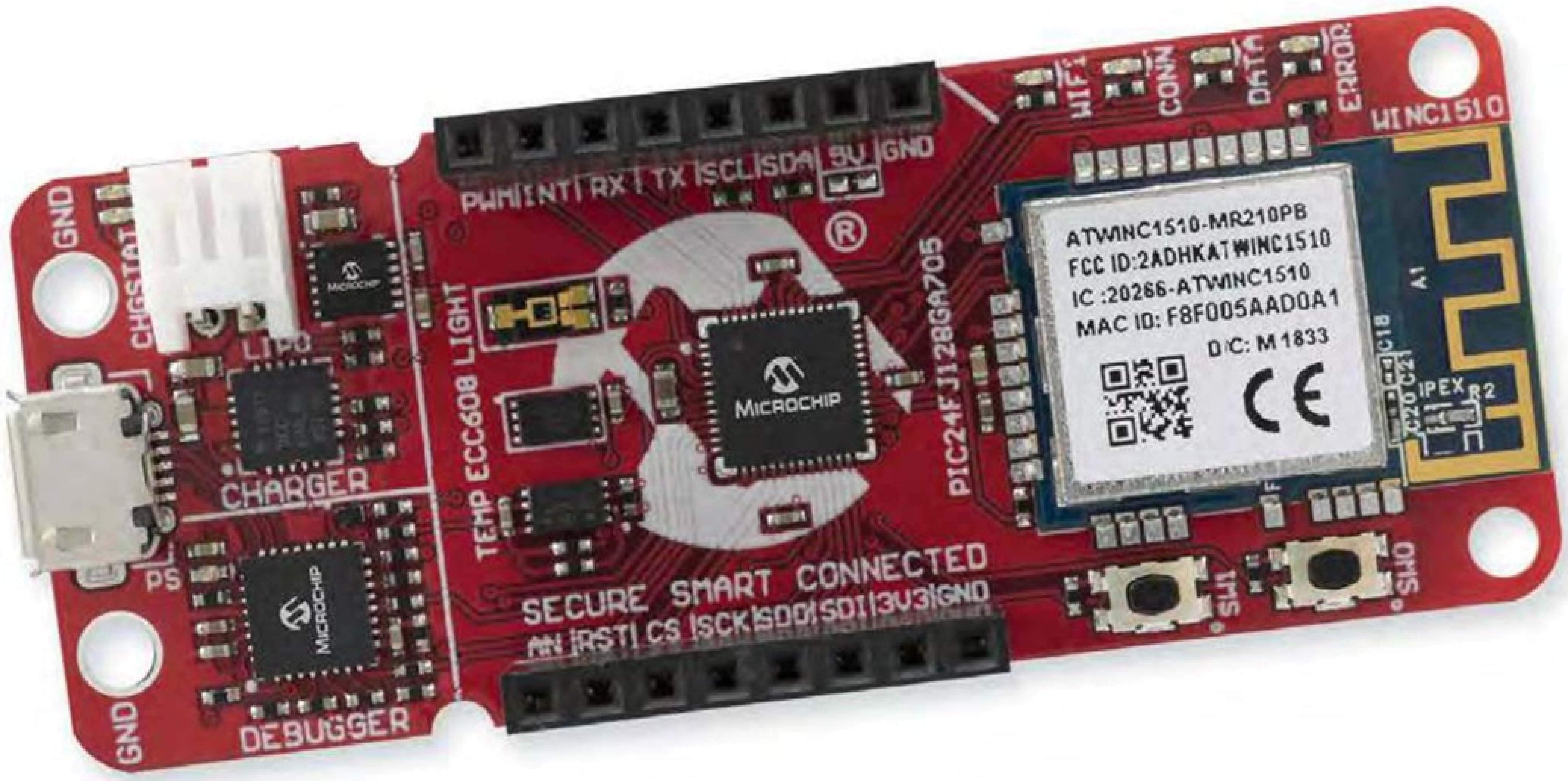
The PIC-IoT WG Development Board combines a powerful PIC24FJ128GA705 MCU, an ATECC608A CryptoAuthentication secure element IC and the fully certified ATWINC1510 Wi-Fi network controller, which provides the most simple and effective way to connect embedded applications to the Google Cloud IoT Core. The board also includes an on-board debugger and requires no external hardware to program and debug the MCU.

Worth $29.00 (approx £23.25) each

The kit board is supported by the MPLAB X Integrated Development Environment (IDE) and MCC rapid prototyping tool. The board is compatible with more than 450 MikroElektronika Click boards that expand sensors and actuator options.

As part of Microchip's extended partnership with Google Cloud, the PIC-IoT WG Development Board enables PIC MCU designers to easily add cloud connectivity to next-generation products using a free online portal at www.PIC-IoT.com. Once connected, developers can use Microchip's MPLAB Code Configurator (MCC) rapid development tool to develop, debug and customise their application.

Free-to-enter competition



Microchip PIC-IoT WG Development Board

How to enter

For your chance to win a Microchip PIC-IoT WG Development Board, enter your details in the online entry form at:

http://page.microchip.com/PE-PIC-IoT-WG-Board.html

Closing date

The closing date for this offer is 31 October 2019.

June 2019 winner

Steve Wilson

He won a Microchip LAN7430 PCIe 3.1 Ethernet Bridge Evaluation Board worth £45.50

ONE FOR MODEL RAILWAY ENTHUSIASTS

DIGITAL COMMAND CONTROL PROGRAMMER for DECODERS

by Tim Blythman

DCC – Digital Command Control – is a widely-used method for controlling model railways, especially when running multiple locos/trains on the same track(s). This DCC programmer is simple, cheap and easy to build – and operates from a computer's USB port.

DCC is a great innovation, allowing many model locomotives to be addressed and operated independently on the same track at the same time. It has been embraced as a standard by the NMRA (National Model Railroad Association, based in the US), so equipment from different manufacturers can inter-operate without issue. And since the standards are public, anybody can create DCC-compatible devices.

But the big downside to DCC (especially for beginners) is the cost of a base station. Even the cheapest base stations cost several hundred pounds and each locomotive also needs a decoder. Usually, even the simplest base stations include the option to program decoders, but this can be a bit fiddly to use and can interfere with the operation of other trains on the 'main line'.

In this article, we describe how to build a standalone programmer based on an Arduino microcontroller module with a custom shield.

It connects to a computer USB port to provide a convenient interface for programming. It allows you to read and write the Configuration Variables (CVs) on DCC rolling stock, customising their operation and performance.

We have also designed the shield to be compatible with the DCC++ Arduino software. DCC++ is an open-source hardware and software system for the operation of DCC-equipped model railroads; see: **https://github.com/DccPlusPlus/**

When programmed with the DCC++ software, the programmer can be used with the Java Model Railroad Interface (JMRI), which provides a way to control DCC-based model trains from a computer. Its most relevant functions for this project are the ability to load and save locomotive (decoder) configurations.

There is also another graphical user interface (GUI) which is compatible with DCC++, written in the 'Processing' language. See the DCC++ web page for details.

As well as being able to use the DCC++ software, we have written a small Arduino program which allows CVs to be read and written via commands on the Arduino Serial Monitor. You don't need JMRI or DCC++ to use it in this mode.

DCC hardware interface

A typical DCC system requires 12-15V to operate (see the two-page end panel for an explanation of how DCC works). You can power an Arduino from a voltage in this range, but it isn't necessary; we're using a small boost regulator module so that you can also run it off a 5V USB supply. In this case, the boost regulator provides the 12-15V DC to power the tracks and DCC decoder(s).

The DCC signal is a square wave at several kilohertz and the locomotive/decoder could draw a few hundred milliamps, so our programmer needs to be capable of rapidly switching the track voltage and supplying sufficient current. Luckily, this can be achieved using a low-cost, standard 556 dual timer IC. This IC is basically two 555 timers together in a 14-pin package.

The outputs on the 556 can deliver up to 200mA at 500kHz, so it is perfectly suitable for this project.

The Arduino module generates the DCC signals with the correct timing, which the 556 converts into the correct voltage levels. We're also using a current sense (shunt) resistor so that the Arduino can detect how much current the attached locomotive is drawing.

The DCC decoder can send data back to the programmer by varying its current draw. It sends a response by briefly drawing at least 60mA from the tracks. This is important, as it is the only way to read data back from the decoder.

Circuit description

Fig.1 shows the circuit of the *DCC Programmer*. It's based around dual 556 timer IC1 and boost converter MOD1.

Pins 4 and 10 of IC1 are the reset inputs of the two timers and the timer outputs are disabled if these pins are low. They are held low initially by the 1kΩ resistor from pin 10 to ground, so output pins 5 and 9 are low. These pins connect to either side of the

Why DCC?

If you're running more than one loco/train on a layout, the only logical way to do it is with DCC, which gives control to each one without affecting any other(s). And because the power and control signals are separate, the locos are still powered even when not moving, so their lights and any sound effects can still be operated.

The downsides of DCC are mostly to do with the cost and there is also the additional complexity of adding decoders to your locomotive.

DCC allows manual control over each train, just as you normally have with a normal controller – but DCC has the big advantage of being able to take its operating commands from a computer, allowing completely automated model layouts, if you wish.

This is especially useful in larger layouts. DCC also allows automatic point switching, level crossing boom gates control and so on, obviously with suitable motors.

The Java Model Railroad Interface (JMRI) software mentioned in the text allows control over all these functions.

Reproduced by arrangement with *SILICON CHIP* magazine 2019.
www.siliconchip.com.au

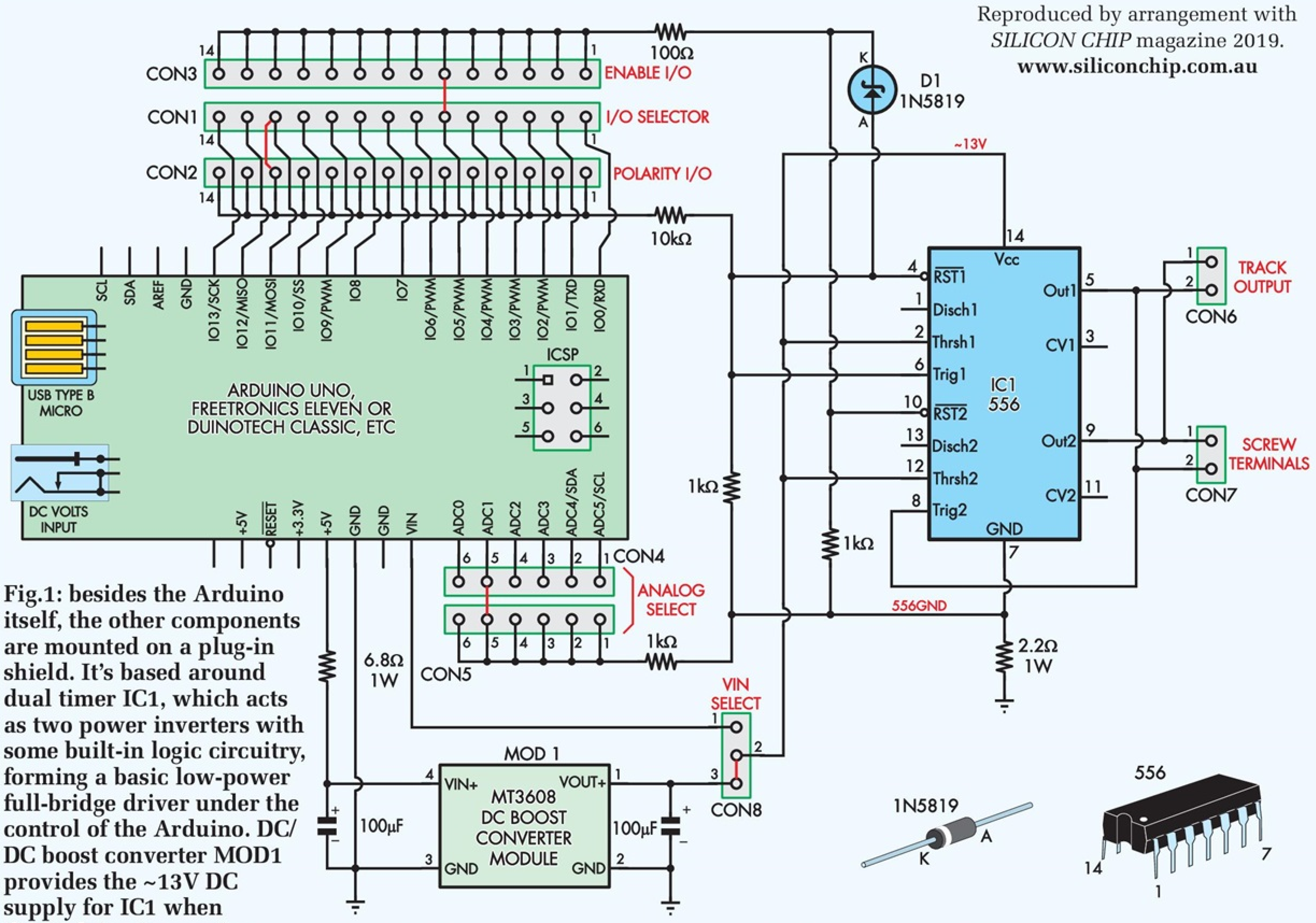

CON3
14
1
ENABLE I/O
100Ω
CON1
I/O SELECTOR
CON2
POLARITY I/O
10kΩ
K
A
D1
1N5819
~13V
SCL
SDA
AREF
GND
IO13/SCK
IO12/MISO
IO11/MOSI
IO10/SS
IO9/PWM
IO8
IO7
IO6/PWM
IO5/PWM
IO4/PWM
IO3/PWM
IO2/PWM
IO1/TXD
IO0/RXD
USB TYPE B MICRO
ICSP
ARDUINO UNO, FREETRONICS ELEVEN OR DUINOTECH CLASSIC, ETC
DC VOLTS INPUT
+5V
RESET
+3.3V
+5V
GND
GND
VIN
ADC0
ADC1
ADC2
ADC3
ADC4/SDA
ADC5/SCL
CON4
ANALOG SELECT
CON5
1kΩ
1kΩ
1kΩ
6.8Ω 1W
Vcc
RST1
Disch1
Thrsh1
Trig1
RST2
Disch2
Thrsh2
Trig2
GND
Out1
CV1
Out2
CV2
IC1 556
TRACK OUTPUT
CON6
SCREW TERMINALS
CON7
556GND
2.2Ω 1W
VIN SELECT
CON8
MOD 1
VIN+
VOUT+
MT3608 DC BOOST CONVERTER MODULE
GND
GND
100µF
100µF
1N5819
K
A
556
14
7
1

Fig.1: besides the Arduino itself, the other components are mounted on a plug-in shield. It's based around dual timer IC1, which acts as two power inverters with some built-in logic circuitry, forming a basic low-power full-bridge driver under the control of the Arduino. DC/DC boost converter MOD1 provides the ~13V DC supply for IC1 when the Arduino is running from a 5V USB supply.

Digital Command Control Decoder Programmer track, via CON6 (a header socket) or CON7 (terminal block), depending on which suits you best.

So when both timers are in reset and both outputs are low, there is no voltage across the track. The enable signal from the Arduino is fed to CON3 so when this goes high, the reset input at pin 10 is pulled high via the 100Ω series resistor.

Schottky diode D1 is then reverse-biased, so that the other reset input at pin 4 can also be driven high by the Arduino, via CON2 and its 10kΩ series resistor.

The trigger input of the first timer (pin 6) is pulled to ground while the threshold input of that same timer (pin 2) is tied to VCC (pin 14), so its output at pin 5 is high by default. However, if the pin 4 reset input is held low by the Arduino (via CON2), then this timer is in its reset state, so output pin 5 is low. Therefore, during operation, the output at pin 5 follows the signal at pin 4.

Output pin 5 is also connected to the trigger input of the second timer, at pin 8, while that timer's threshold input (pin 12) is also tied to VCC. So output pin 9 is low when output pin 5 is high and vice versa, and thus the second timer operates as an inverter once it is enabled.

Thus, when IC1 is enabled, there is always a voltage across the track, with the magnitude being close to VCC and the polarity depending on the signal from CON2.

A 2.2Ω resistor between IC1's ground pin and the main circuit ground is used to sense the current flowing between the track connections.

Irrespective of the direction of current flow through the track connections, that current must ultimately always flow back through IC1 and then to the power supply ground. Therefore, the result is a voltage across that 2.2Ω resistor.

That voltage is fed to CON5 via a 1kΩ series resistor and on to one of the Arduino's analogue, pins where it feeds the internal 10-bit analogue-to-digital converter (ADC) and is converted into a digital value in the range of 0-1023 by the software. The combination of the 10-bit ADC and the 2.2Ω current-sense resistor means the resolution of this reading is around 2mA.

JP1 is a three-pin header which allows you to choose whether the voltage fed to the 556 IC is from the Arduino's DC input socket (via the VIN pin) or from MOD1, an MT3608 2A boost DC/DC converter module. If you are powering the Arduino from a 12V plugpack, then you can put JP1 in the VIN position, so the plugpack provides the track voltage.

But if you are powering the Arduino from a computer or a 5V USB charger, this voltage is not sufficient to power the tracks. In this case, you can place JP1 in the VOUT+ position and adjust the trimpot on the MT3608 module to provide 13V to IC1.

While not vital, as MOD1 has onboard bypass capacitors, the shield has provision for 100µF input bypass and output filter capacitors. We recommend that you fit the input bypass capacitor, but leave the output filter capacitor off (as described in the *Construction* section).

There is also an onboard 6.8Ω resistor between the 5V supply and the module's VIN+ terminal, to reduce the inrush current from the 5V supply when it is first connected and to limit the maximum current drawn during operation (eg, if the tracks are accidentally shorted).

The shield is driven using two of the Arduino's digital pins and feeds the current-sense voltage back to one of the analogue pins. Rather than make these fixed, and risk interfering with the function of any other shields plugged into the Arduino, all three pins can be selected using jumper shunts for maximum flexibility.

The digital pins are selected by placing one jumper between CON1 and CON2 (for the polarity signal) and another between CON1 and CON3 (for the enable signal). By default, our software is set up to use digital output D5 for the polarity signal and D11 for the enable signal, but you can change the software if you place the jumpers in other locations.

Similarly, the analogue pin is selected by connecting a pin on CON4 to the associated pin on CON5 and by default, we're using analogue input A1.

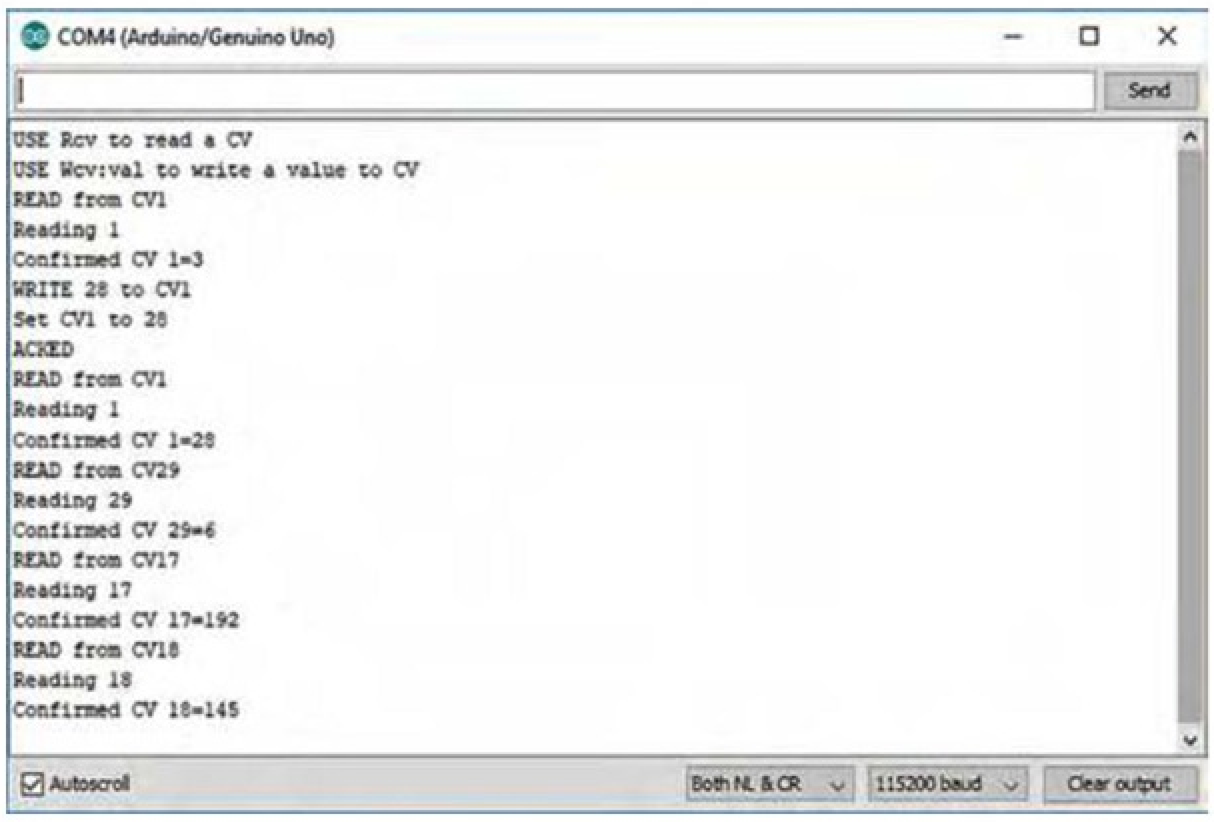
The software

The primary job of the software is to produce a signal across the tracks which carries the required DCC packets. It does this using an interrupt software routine (ISR) which is triggered every 58µs by a timer. 58µs is the half-period of a '1' bit.

To send a '1' bit, we toggle the DIR pin each time the interrupt fires. To send a '0', we should ideally have a half-period of 100µs, but 116µs (2 x 58µs) is within the limits that are recognised by the decoder. So we merely wait for two timer interrupts to occur before toggling the DIR pin to transmit a zero bit.

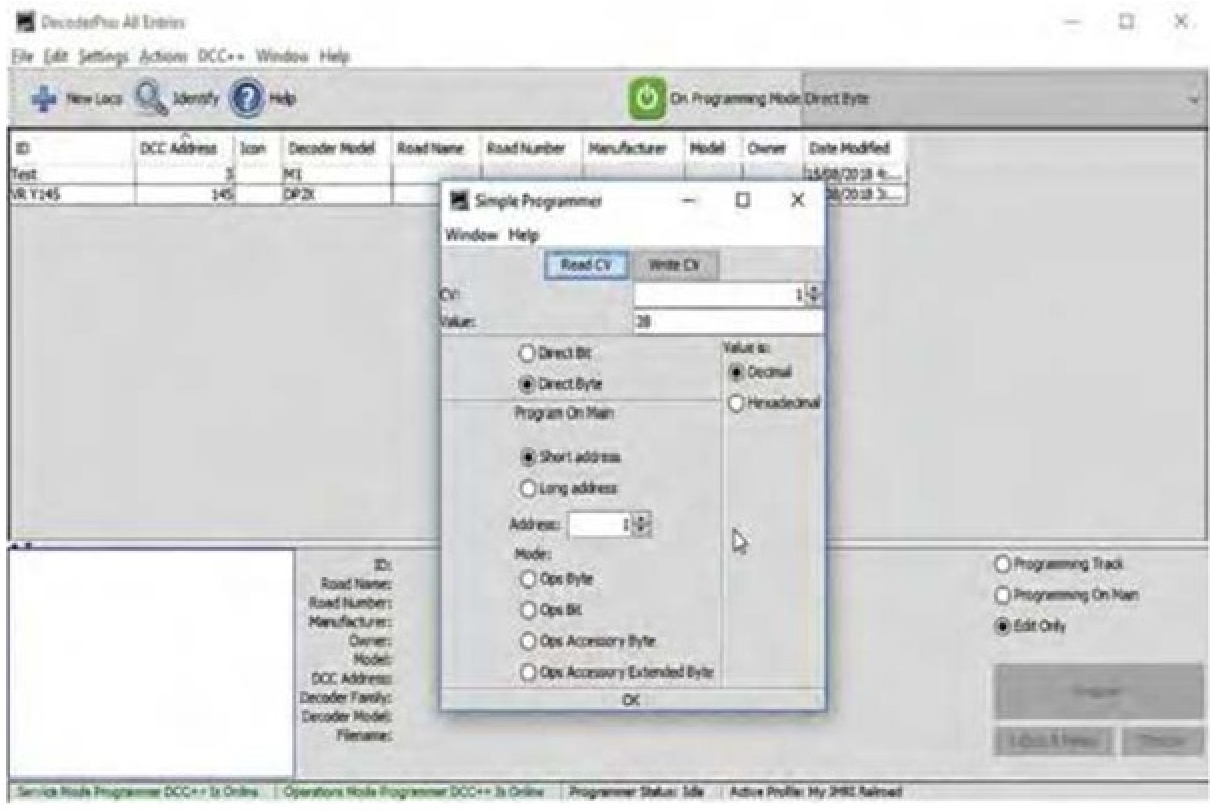
The interrupt handler steps through the DCC data array as each bit is transmitted, then sets a flag to indicate that the complete packet has been sent and moves onto the next packet. Thus, most of the real work is done inside the interrupt routine.

During most programming sequences, the programmer sends several reset packets to the loco, followed by



```
USE Rcv to read a CV
USE Wcv:val to write a value to CV
READ from CV1
Reading 1
Confirmed CV 1=3
WRITE 28 to CV1
Set CV1 to 28
ACKED
READ from CV1
Reading 1
Confirmed CV 1=28
READ from CV29
Reading 29
Confirmed CV 29=6
READ from CV17
Reading 17
Confirmed CV 17=192
READ from CV18
Reading 18
Confirmed CV 18=145
```

The stand-alone Arduino program has a very simple interface, and allows direct reading and writing of CVs. This is sufficient to fully manipulate any parameters, but may not be as intuitive as the advanced roster settings available with DecoderPro.



ID	DCC Address	Icon	Decoder Model	Road Name	Road Number	Manufacturer	Model	Owner	Date Modified
Test	3		M1						14/08/2018 4...
VR Y145	145		DP2X						28/2018 3...

CV: 1
Value: 28

○ Direct Bit
● Direct Byte

Value is:
● Decimal
○ Hexadecimal

Program On Main
● Short address
○ Long address
Address: 1
Mode:
○ Ops Byte
○ Ops Bit
○ Ops Accessory Byte
○ Ops Accessory Extended Byte

○ Programming Track
○ Programming On Main
● Edit Only

Service Mode Programmer DCC++ Is Online | Operations Mode Programmer DCC++ Is Online | Programmer Status: Idle | Active Profile: My JMRI Railroad

This is the Simple CV Programmer interface in DecoderPro, and it allows CVs to be directly written and read. Using the Roster feature allows the locomotive to be given a name, and CVs to be saved to a configuration file, as well as grouping the CVs into logical groups.

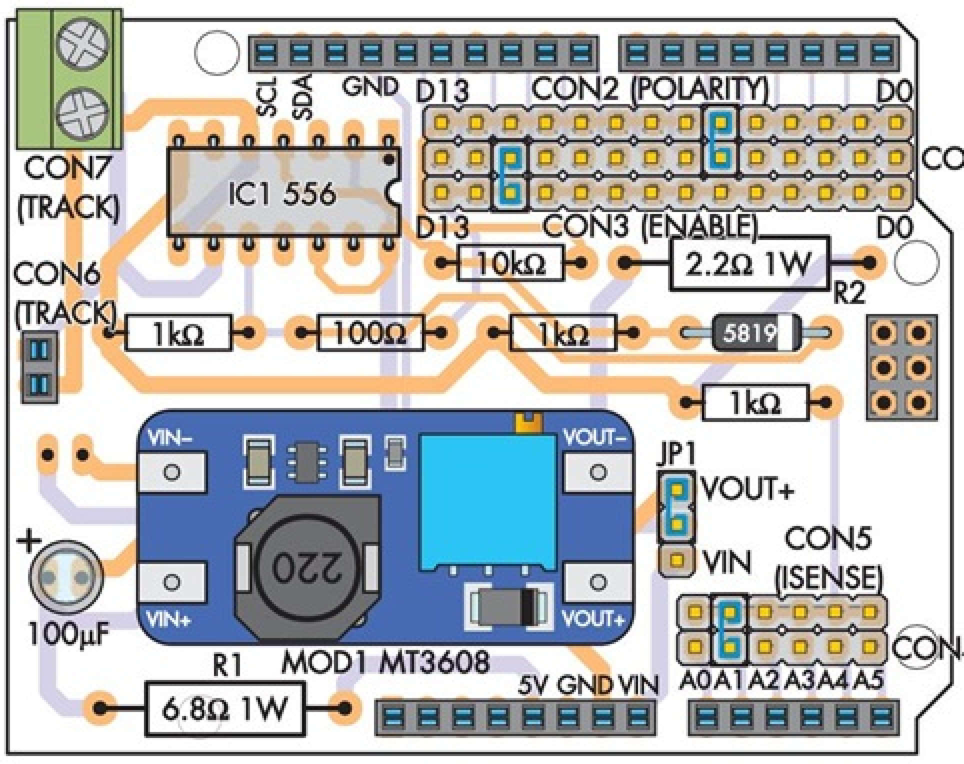
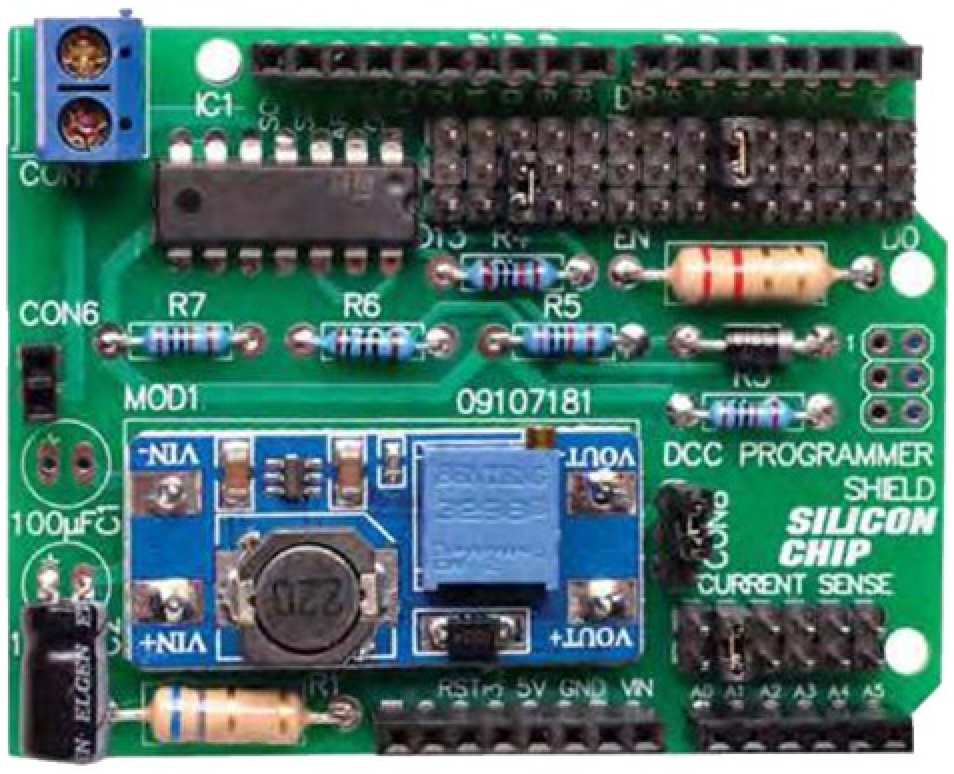


Fig.2: use this PCB overlay diagram and the photo at right as a guide when building the shield. Make sure that MOD1 is oriented correctly, with its VIN pins towards the electrolytic capacitor. IC1, D1 and the electrolytic capacitor are also polarised. The jumper locations shown here suit our sketch as well as the DCC++ code, when being used as a DCC programmer.



multiple 'write' packets, and this is accomplished by placing the appropriate commands in the array to be transmitted.

The detection of acknowledgement pulses from the loco is done by continually sampling the voltage across the current sense resistor, at the selected Arduino analogue input. The quiescent sample value is used as a baseline value. If an acknowledgement is expected, the current sample is compared with the baseline and if the threshold is reached, a flag is set indicating an acknowledgement has been received.

As well as being able to use the DCC++ software, we provide a basic serial terminal interface. This allows you to send DCC commands straight to the locomotive, assuming that you know what is required. These are inserted into the array of packets to be sent, but they are not transmitted until you indicate that they are ready. The software then sets a flag to start the transmission.

MOD1 is optional

We are using boost regulator MOD1 for convenience, so that you can run the programmer off a 5V USB supply. However, if you plan to power the Arduino from a 12-15V DC supply, you could omit MOD1. In this case, you would need to place JP1 in the 'VIN' position. Keep that in mind as you assemble the shield.

Construction

The *DCC Programmer* Shield is built on a PCB measuring 68.5 x 53.5mm (the size and shape of a standard Arduino shield) and coded 09107181, available from the *PE PCB Service* – see page 78. Use the overlay diagram in Fig.2 as a guide during construction.

Fit the smaller resistors first, confirming the values with a DMM before soldering each in place. Follow with diode D1, taking care to orient it as shown in Fig.2. Then install the two larger 1W resistors.

You can now fit the 100µF input bypass capacitor for MOD1. This should be laid over on the PCB so that you can later stack another shield on top if you need to. Make sure its positive (longer) lead goes in the pad nearest the adjacent edge of the PCB.

Next, solder in pin headers CON1-CON6 and JP1 where shown. You can then fit MOD1, by first soldering four component lead off-cuts to the pads so that they stick out the top of the board, then lowering MOD1 over those leads and soldering them to the pads on the top of the module. You can then trim off the excess lead length on top of MOD1 and on the underside of the PCB.

Now solder CON7 to the board, making sure it has been pushed down so it is sitting correctly on top of the PCB and its wire entry holes are facing towards the outside edge of the board.

Fit the Arduino headers next. We've specified stackable headers but you could potentially use standard headers if you don't plan to attach any shields on top of this board. Either way, make sure the long pins project out the bottom of the board. You need to solder the stackable headers carefully to avoid getting solder on the pins except near where they connect to the pads.

Note that the stackable headers give more clearance for the components on the board underneath (eg, the Arduino). If you use standard headers, you may find that CON7's pads short to the shield of the USB connector below, which is connected to ground.

Adjusting MOD1's output voltage

Before installing IC1, you must adjust MOD1's output voltage. Plug the shield into your Arduino board and then apply power.

Use a DMM set to measure DC volts to probe the VOUT+ and VOUT– pads on MOD1. Adjust its onboard trimpot screw to get a reading just below 15V; counter-intuitively, the voltage is decreased by turning the adjustment screw clockwise. We set our track voltage to 13V, and it works well.

Parts list – Arduino DCC Decoder Programmer

1 double-sided PCB, code 09107181, 68.5 x 53.5m
1 set of Arduino stackable headers (1 x 6-pin, 2 x 8-pin, 1 x 10-pin)
1 Arduino Uno or equivalent
1 MT3608 2A boost module [Banggood ID: 1213134]
3 14-pin headers (CON1-CON3)
2 6-pin headers (CON4,CON5)
1 2-pin female header socket (CON6)
1 2-way terminal block, 5/5.08mm pin spacing (CON7)
1 3-way pin header (JP1)
4 jumper shunts/shorting blocks

Semiconductors
1 556 dual timer IC (IC1)
1 1N5819 1A schottky diode (D1)

Capacitors
1 100µF 25V electrolytic

Resistors (all 0.25W, 1% unless otherwise stated)
1 10kΩ   3 1kΩ   1 100Ω   1 6.8Ω 1W 5%   1 2.2Ω 1W 5%

You can then remove power, unplug the shield and fit IC1, ensuring that its pin 1 notch faces into the middle of the board, as shown in Fig.2. You could potentially use a socket, but given that the IC is supplying significant current to the tracks, it's better to avoid that.

Now fit the jumpers to select your desired track voltage source and to configure which Arduino pins are used for control. If you are unsure, insert the jumpers in the positions shown in Fig.2. The shield assembly is then complete.

Software setup and testing

Now you need to upload the Arduino sketch code to the board. This is done using the free Arduino IDE (integrated development environment). It is available for Windows, macOS and Linux. The IDE is used to compile and upload the software to the Arduino. You can download this software from: **www.arduino.cc/en/Main/Software**

Now download the software package from the October 2019 page of the *PE* website. This contains a 'standalone' Arduino sketch called **DCC_Programmer_Shield_V2**, which is the easiest way to test your shield. Before you can use this sketch, you need to install a library called **TimerOne**.

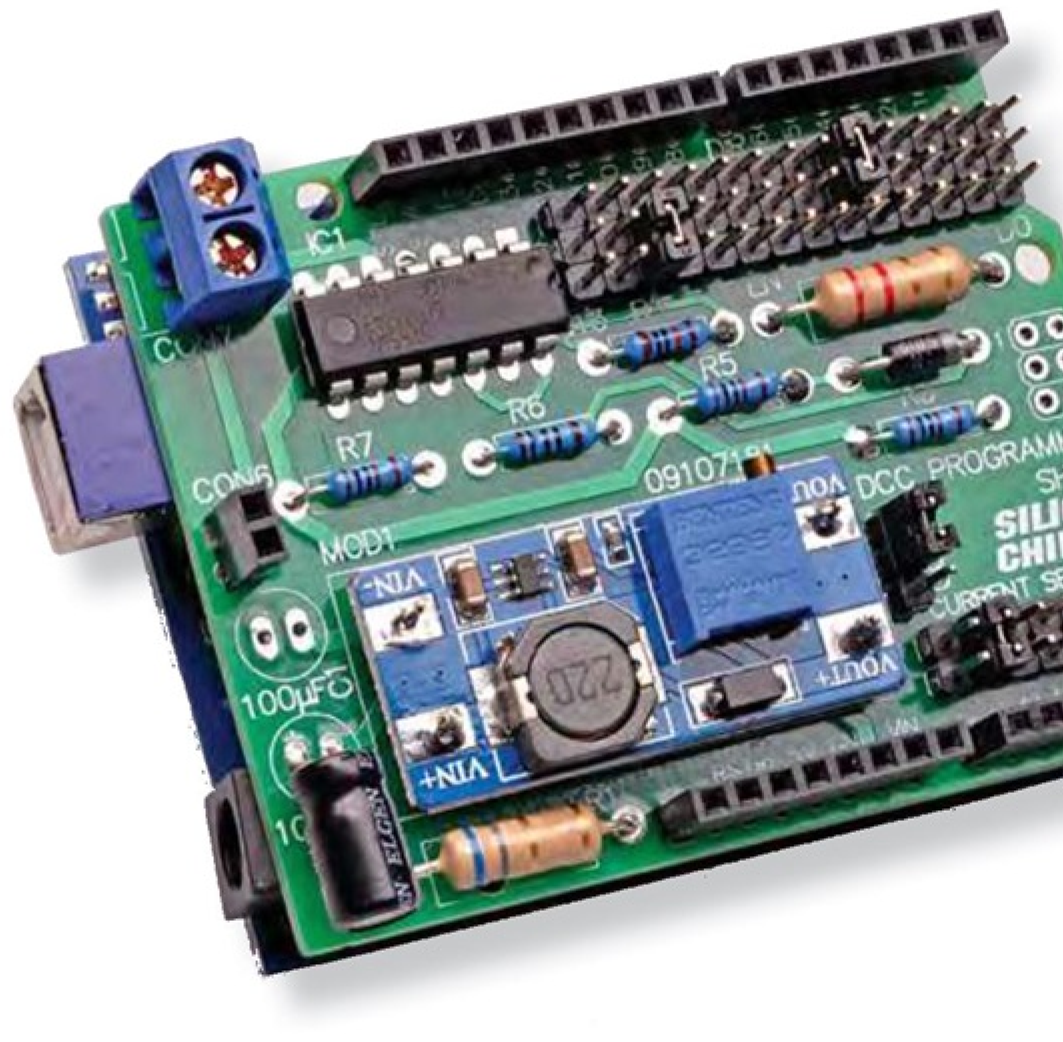
TimerOne can be installed from the IDE's Library Manager (Sketch menu → Include Library → Manage Libraries…). Open the Library Manager and search for **TimerOne** and then when you find it, select it and click the 'Install' button.

In case you have trouble with that method, we also include a zipped copy of the library in the sketch download package. This can be installed using the Sketch → Include Library → Add .ZIP Library menu option.

Now extract and open the **DCC_Programmer_Shield_V2** sketch from the download package; eg, using the IDE's File → Open menu option. Connect the Uno to your computer and select the 'Uno/Genuino' option from the Tools → Board menu.

Also, check that the correct COM port listed under the Tools → Port menu matches the one assigned to your Uno board.

Select the Upload option under the Sketch menu and then open the serial monitor (Tools → Serial Monitor) at 115,200 baud. You should get a message describing how to use the sketch. If you have a DCC locomotive, place it on a length of track wired to either CON6 or CON7 and type 'r1' in the Serial Monitor, followed by the Enter key.



The DCC programmer shield mounts on top of the Arduino UNO board, as seen here.

If everything is working, you should see some text appear on the Serial Monitor indicating the locomotive's short address (CV 1). Other CVs can be written to and read from with simple commands like this. To write a CV, use the format **w1:3** That command writes the value 3 to CV 1.

Using it with DCC++ and JMRI DecoderPro

If you want to run DCC trains from a computer, you can use the open-source JMRI software. It comes with DecoderPro, which has a comprehensive 'Roster' feature which can be used to save and restore locomotive programming parameters.

If you have a fleet of similar locomotives, this is a convenient way of managing their various performance CVs.

So that our Programmer can operate with JMRI, we can use the open-source DCC++ Arduino sketch. This was designed to work with other hardware, but our shield has been designed to be compatible with that hardware, so you can run DCC++ on it without making any modifications.

Our software download package for this project includes the DCC++ software. To use it, extract and open the **DCCpp_uno sketch**. Then check the 'Config.h' tab and make sure that this line is set correctly:

```
#define MOTOR_SHIELD_TYPE 0
```

You need to set up the jumpers on CON1-CON5 for your shield to match those shown in the overlay diagram in Fig.2. This configuration is required to emulate this motor shield type. Upload the sketch to the Uno board, as described above.

The DCC++ software also uses a serial interface, so you can use the serial monitor to examine its output.

The DecoderPro program has the option to use a DCC++ programmed Uno, so our reprogrammed board can now be used with DecoderPro. JMRI is available for Windows, Mac and Linux (including a Raspberry Pi version). However, note that you need to have Java installed to use it.

Now, get JMRI from: **http://jmri.sourceforge.net/download/index.shtml** and open the DecoderPro program. Go to the Edit → Preferences menu and under Connections, choose DCC++ as System Manufacturer and DCC++ Serial Port as System connection. Ensure the Serial port matches that of the Uno.

Save the configuration and close DecoderPro so that it can reload the new settings. Open DecoderPro again and under Edit → Preferences choose defaults, and ensure that DCC++ is selected for Service Programmer. Unless you have other hardware, you should select DCC++ for all options.

Save, close and reopen DecoderPro again. Click the red power button and ensure that it turns green. You should see 'Service Mode Programmer DCC++ Is Online' in the bottom left corner of the screen.

Now use the Actions → Single CV Programmer menu option to open the Simple Programmer window. This window allows you to read and write single CVs via a basic interface.

If you find you are getting the 'No acknowledge from locomotive' error then the threshold that DCC++ uses for detecting acknowledge pulses from the track may need to be adjusted. This is a parameter in the **DCCpp_uno** sketch.

Close DecoderPro (so that it no longer has control over the Arduino) and navigate to the 'PacketRegister.h' tab in the **DCCpp_uno** sketch. Near line 20, there is a value called ACK_SAMPLE_THRESHOLD which is 30 by default. We found that reducing this to 12 gave consistent results with three different decoders.

Save the sketch with the changes, then upload it to the Arduino board again. Ensure that the Serial Monitor is closed before reopening DecoderPro and try the Single CV Programmer option again.

While the default value of 30 corresponds to the specified value of 60mA for the acknowledge pulse,

FD819MX7

DCC
ARDUINO
DCC PROGRAMMER SHIELD
SILICON CHIP
CURRENT SENSE

OVERLEAF
HOW DCC WORKS

the DCC++ sketch also applies some smoothing to the sensed current, so this may change the actual detected current threshold.

To use the Roster feature, click the 'New Loco' button in the top left corner of the main DecoderPro window. Click 'Read type from decoder' and DecoderPro will read a number of the CVs to identify parameters such as the manufacturer. If this does not work, you can also choose Roster → Create entry to enter this manually.

Once a Roster entry has been created, double-click it to open the Comprehensive Programmer, allowing more detailed and complex programming to occur.

JMRI is a vast program with many features and we can't pretend that we've covered a fraction of them here. There is comprehensive documentation online.

The article at: **http://bit.ly/pe-oct19-decpro** introduces DecoderPro and is a good place to start.

JMRI also has a layout editor (PanelPro) so that you can create track diagrams and these can be animated with information from the layout if you have the correct sensors installed.

Other uses for this project

You may not think that having a *DCC Programmer* is all that useful, but this device can also be used as a minimal DCC base station. With the DCC++ software, the Programmer can act as a low-power booster.

If you set the board jumpers for EN on pin D3, DIR on pin D10 and Current Sense on A0 then the DCC++ sketch can use the Programmer as its 'main line' output.

This means that the output can be connected to some track and DecoderPro's throttle window can be used to control a DCC decoder-equipped locomotive.

Because its current capacity is limited, you're not going to be running a fleet of trains, but it could be handy for testing and experimentation.

In our trials, we found a small H0-scale locomotive was able to be driven at low speed, including operating the lights.

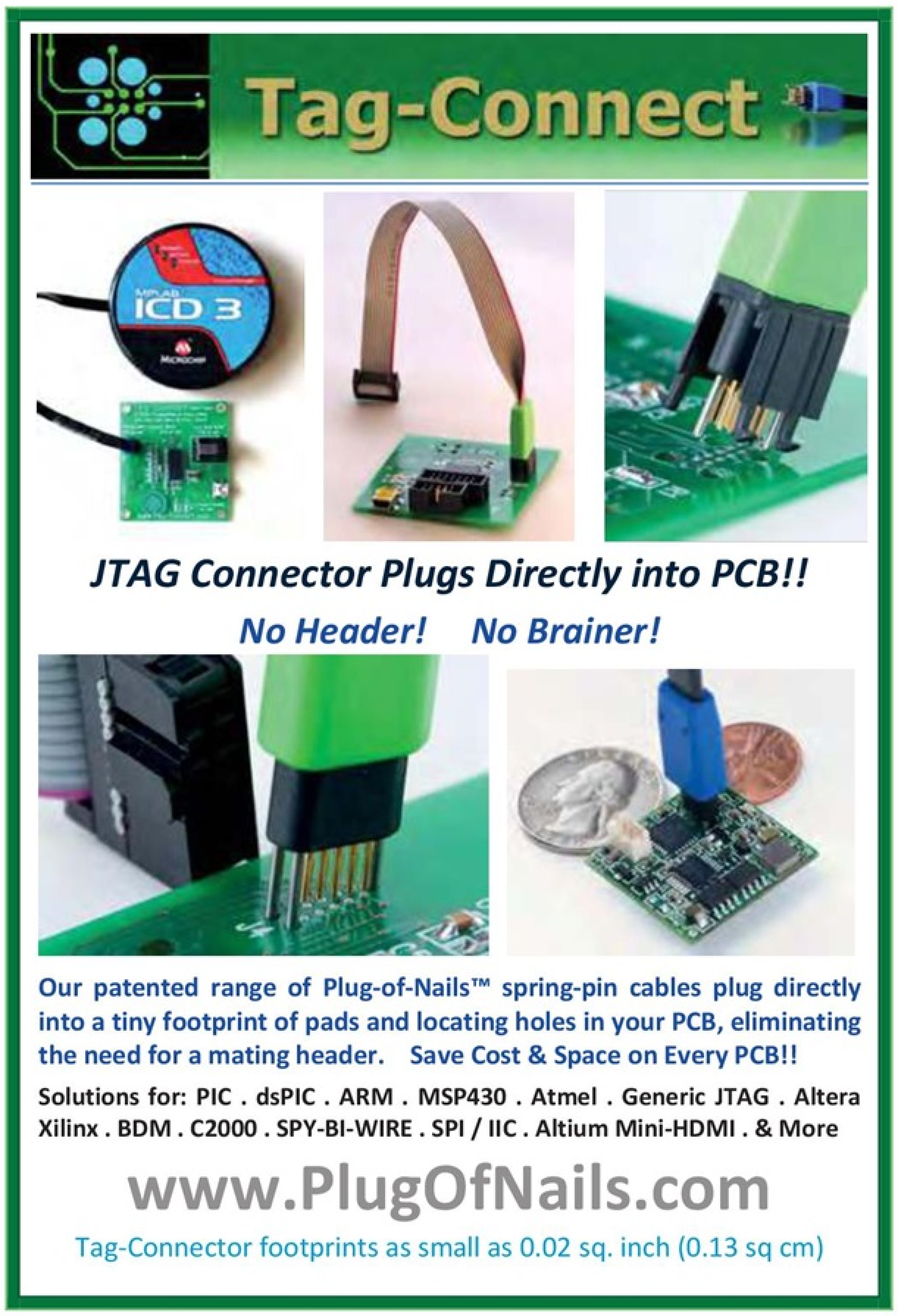
What about driving a small motor?

Another possible use for this shield is as a small 12V motor-driver board. If you need to drive a small 6-15V DC motor using USB power or some other source of 5V then this shield can be used as a reversible motor driver, without the need for an external power source. It's suitable for motors drawing up to about 2W. Note that your 5V supply must be able to deliver enough current.

Useful links

Setting up DCC++ with JMRI DecoderPro:
https://github.com/DccPlusPlus/BaseStation

DCC Standards Page:
http://bit.ly/pe-oct19-nmra


Tag-Connect

JTAG Connector Plugs Directly into PCB!!

No Header! No Brainer!

Our patented range of Plug-of-Nails™ spring-pin cables plug directly into a tiny footprint of pads and locating holes in your PCB, eliminating the need for a mating header. Save Cost & Space on Every PCB!!

Solutions for: PIC . dsPIC . ARM . MSP430 . Atmel . Generic JTAG . Altera Xilinx . BDM . C2000 . SPY-BI-WIRE . SPI / IIC . Altium Mini-HDMI . & More

www.PlugOfNails.com

Tag-Connector footprints as small as 0.02 sq. inch (0.13 sq cm)

How DCC works

If you are not a model railway enthusiast, DCC's advantages may not be obvious. To understand why it's so useful, we'll explain how model railway systems worked before DCC.

For a standard DC model railway, a single locomotive is fed power through the tracks, with one rail being negative and the other positive. Varying the average voltage between the rails changes the loco's speed, while swapping the rail polarities reverses its direction of travel. A typical H0-scale locomotive (1:87 scale) runs from 12V DC, drawing from a few hundred milliamps up to an amp or so.

Such a system only allows a single locomotive to be controlled, as multiple locomotives would receive the same track voltage. And direct control of accessories such as headlights, steam or sound effects is not possible. With clever use of diodes, it's possible to have directional headlights, but a battery is still needed for lights if the locomotive is stopped, as there is no voltage on the track.

The most common way to allow multiple locomotives to operate is to divide the track into individual 'blocks' or sub-circuits, which can be switched between controllers. As you can imagine, the switching rapidly becomes very complicated as the track layout expands or more controllers are added.

DCC transmits both power and control signals using the tracks. The locomotive's onboard decoder interprets the control signals and commands the motors, lights and other accessories. Power is supplied continuously, allowing accessories to operate even while the locomotive is stationary. Beyond lights, features such as smoke generators and sound effects modules are quite common (if not especially cheap).

The DCC signal is a square wave with a 50% duty cycle and varying pulse width. The data is encoded in the pulse widths, while the square wave, once rectified, provides DC power for the decoder, motor, lights and so on.

The track voltage is typically 30V peak-to-peak, with cycle times of around 100µs. A typical DCC base station consists of a microcontroller feeding an H-bridge of some sort, usually with some form of current detection to shut it down in the event of a short circuit. With exposed conductors in the form of rails, it's bound to happen sooner or later.

A simplified circuit diagram of a typical DCC decoder is shown in Fig.3. This shows some of the circuitry to drive two lamps (eg, headlight and marker LEDs) as well as the motor.

The DCC protocol encodes a binary 1 as a pulse that is nominally 58µs high and 58µs low, with a binary 0 having high and low periods that are 100µs or longer (see Fig.4). Because the locomotive can be placed on the track either forwards or backwards, the absolute polarity cannot be determined, so the high pulse may come before or after the low pulse.

Fig.5 shows what these '0' and '1' bit pulses look like when they are arranged back-to-back, forming a continuous AC waveform.

While the standards are specific as to what should and should not be accepted as valid data, merely sampling the 'high' period of adjacent bits is sufficient to decode the bitstream, and in any case, each command has a checksum byte, so errors caused by timing inaccuracies can be detected and the corrupt command ignored.

The peak frequency of this signal is around 8kHz, which is higher than the 5kHz that the widely-used L293 full-bridge motor driver IC can deliver. So DCC base stations designed around the L293 IC probably won't work reliably.

DCC protocol

The bitstream is broken up into packets, with each packet containing one command, for tasks such as setting the motor to a particular speed or turning a light on and off. The loco's state is stored by the decoder and kept consistent until it is updated by a future packet.

There are also dedicated programming commands that set configuration values (CVs) in non-volatile EEPROM. These values are used to determine which address the locomotive responds to, how the speed changes with throttle position and how lights and other accessories react to inputs. It's these CVs that our programmer is designed to read and modify.

Each valid packet starts with at least 14 sequential '1' bits. This is referred to as the 'preamble', and allows the decoder to synchronise itself with the start of the data packet, which is indicated by the first following '0' bit. The data for the first byte of the command (which is an address) follows this, with all bytes sent most-significant-bit (MSB) first (see Fig.6).

Subsequent bytes are prefixed with a zero bit, so each byte transfer requires nine bits to be sent. The last byte to be transmitted is the aforementioned checksum byte.

After this, a '1' bit is sent, indicating that the packet is complete. The final '1' can also count as the start of

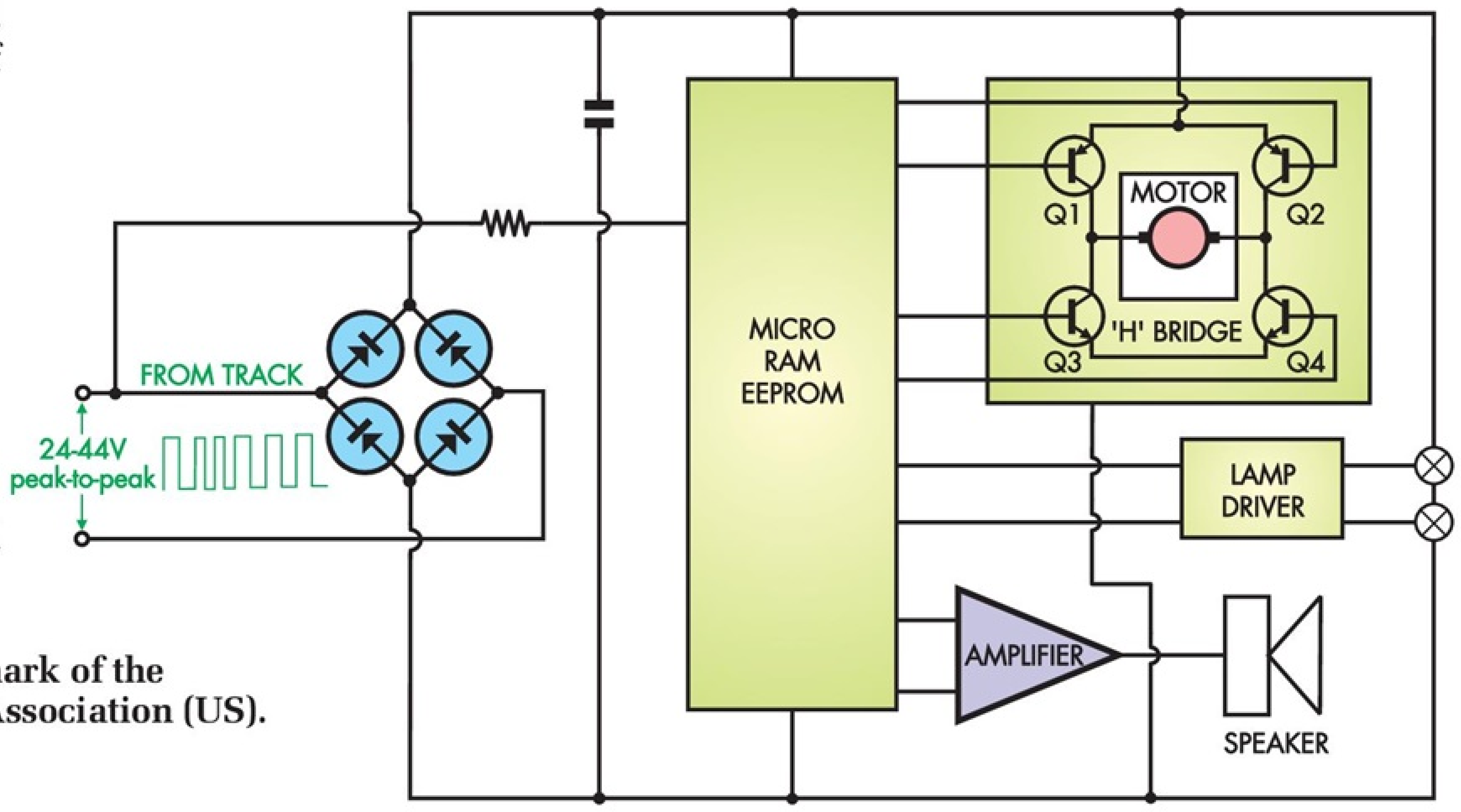


Fig.3: a simplified version of a typical DCC decoder circuit, showing the bridge rectifier and filter capacitor used to convert the AC voltage on the tracks to a DC voltage, with the raw track signal also being fed to the micro so it can be decoded. The micro then controls the H-bridge motor driver transistors, lamps, sound effects module and other accessories.

® – Registered DCC trademark of the National Model Railroad Association (US).

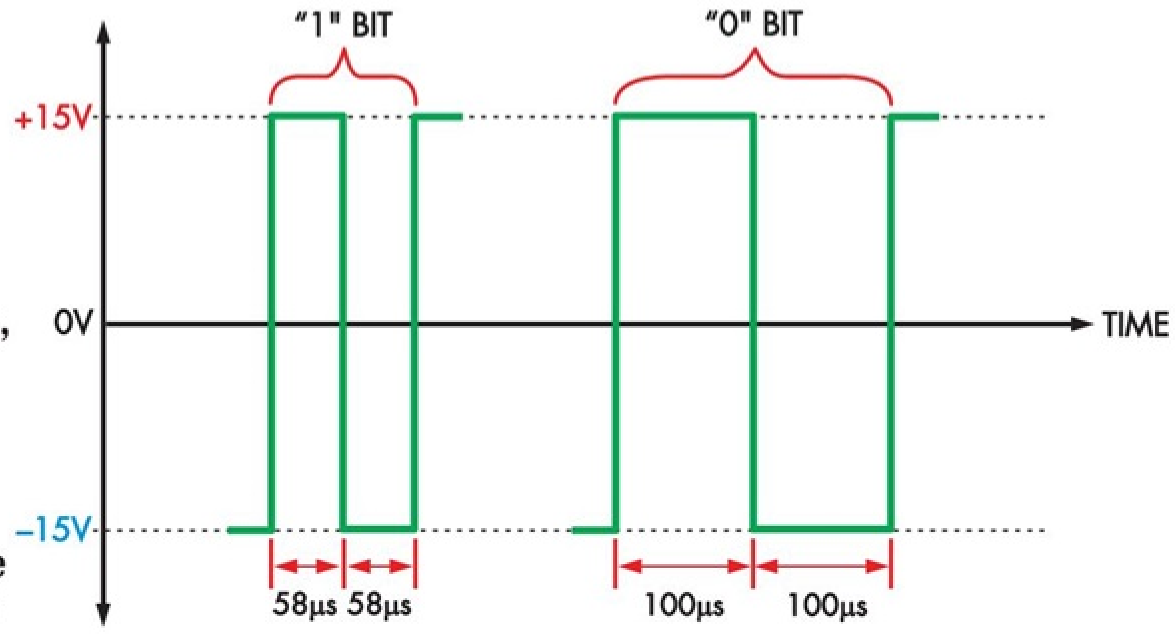


Fig.4: in a DCC system, the voltage across the tracks has a more-or-less constant magnitude, but its polarity is continuously reversing, creating a 5-8kHz square wave. This is rectified by the DCC decoder(s) and is used as a power source for their logic circuitry as well as driving motors, lights, audio amplifiers... but the frequency modulation also encodes digital commands, addressed to individual locos.

the next preamble if packets are sent back-to-back.

Since each byte in the packet is separated by a '0' bit, the only place that more than eight '1' bits can appear in a row in a valid DCC sequence is in the preamble, so decoders can't be fooled into thinking a new packet is starting in the middle of a valid packet.

Each packet is a minimum of three bytes, so the minimum packet transmission length is 40 bits, giving a minimum packet time of just over 5ms. So it's possible to send close to 200 commands per second using DCC.

Once the decoder has received a packet and the checksum is correct, it checks the address byte. If it matches the address stored in the decoder's EEPROM, it can act on the command and if necessary, send a response. That is generally only necessary for programming commands, as the programmer needs a way to read the configuration from the decoder. In this case, the packet sent by the base station is equivalent to asking 'is this bit of this configuration variable set?', to which the decoder replies either with an acknowledgement or not.

The acknowledgement is performed by the decoder by placing a load of 60mA or more across the tracks for 6ms. In practice, this is usually done by the decoder briefly powering the motor, often resulting in the locomotive inching forwards during programming.

The acknowledgement is one of two reasons why programming usually occurs on a dedicated programming track. First, it would be difficult to accurately detect the acknowledgement pulse in the presence of other loads (such as other locomotives) on the rails.

Second, many programming packets are broadcast to all decoders on the track. So the only way to guarantee that only the correct locomotive receives the programming packets is to have a dedicated section of track for programming.

The DCC standards are very detailed and make for interesting reading. It could potentially be used in situations other than model railways, where power and commands need to be sent and received over two wires.

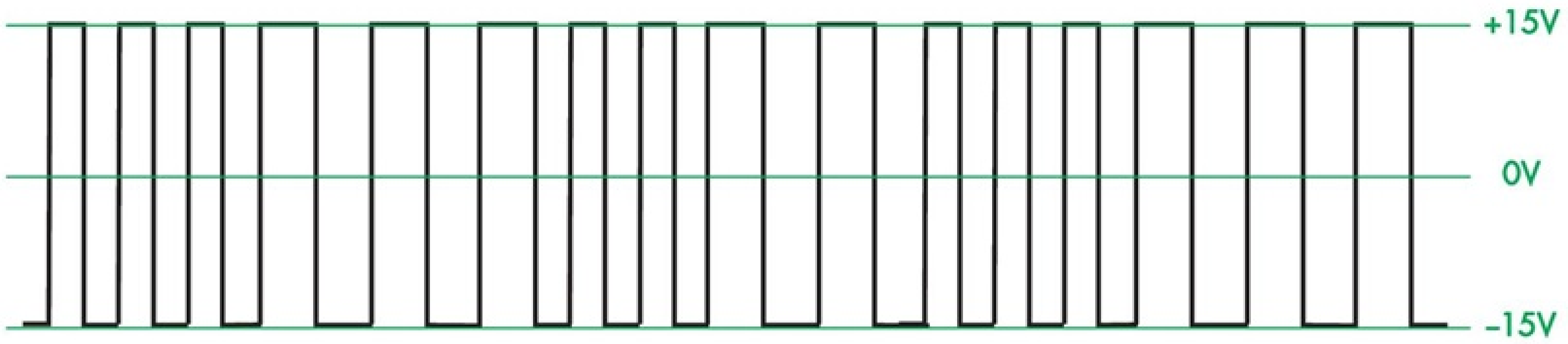


Fig.5: the sequences of '0' and '1' bit pulses are strung together to create a continuous AC waveform across the tracks, effectively forming a frequency-modulated square wave. The RMS voltage is very close to the peak voltage, providing a similar voltage and current to the motors in the locos, while the average voltage is effectively zero since the waveform is symmetrical.

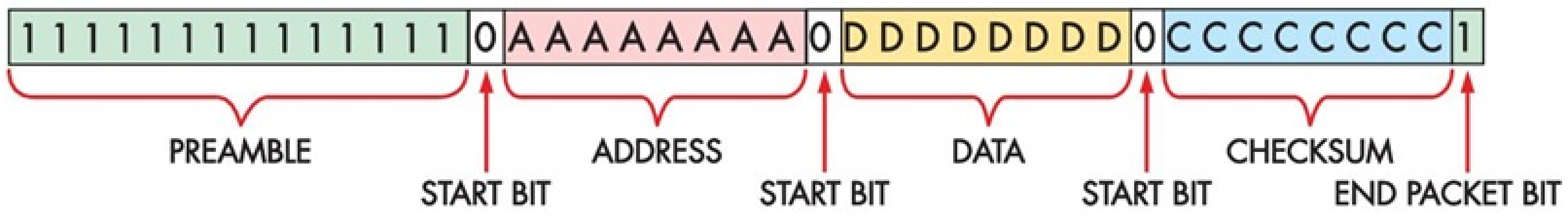
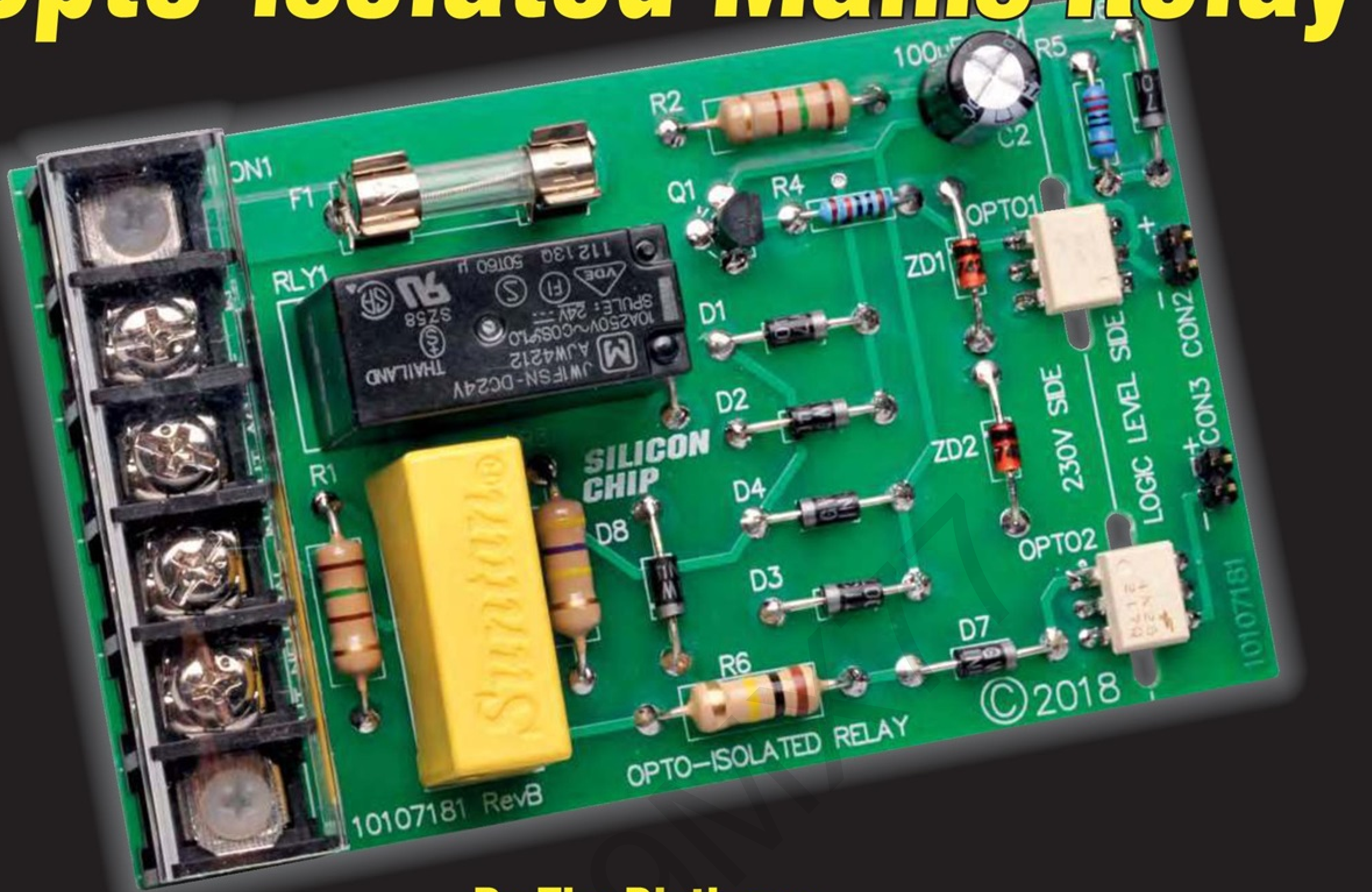


Fig.6: this illustrates the structure of the shortest possible valid DCC command, containing one address and one data (command) byte. The preamble always consists of at least fourteen sequential '1' bits and each byte is separated by a '0' bit, with the command being terminated by a checksum byte (to detect errors) and then a final '1' bit to indicate that there is no further data in this command.

PoLabs
For more information or software download please visit
www.poscope.com/epe
PoKeys
Connect, Control
- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC
Stepper motor drivers
Drive
- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A
PoScope Mega1+
PoScope Mega50
Measure
- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

Keep the bitey bits out of harm's way!

Opto-isolated Mains Relay



By Tim Blythman

If you need to switch mains voltages, say from a micro's output or any other low-voltage source, you need to isolate them from each other. That's what this project does – it's easy to build and keeps mains voltages locked away from the controller... and you! Virtually any low-voltage source will do – from 2.7 to 10V.

Arduino and Raspberry Pi modules are popular because it's so easy to get into them, even if you're a beginner. But many people do not like working with mains, and with good reason – it's easy to create an unsafe situation if you don't know better.

Incorrect wiring or inadequate insulation is a hazard, not just to you, but to anyone who comes in contact with your invention.

This project is an ideal way of switching mains power, whether you are a beginner or not. If you follow the instructions in this article carefully, within an hour or two, you will have a working – and more importantly – safe, mains switch.

You could control a heater, light, fan, pump, television, amplifier, computer – just about anything that plugs into a mains socket.

You can use a wide range of sensors to decide when to switch those devices on and off; we've covered many easy-to-use sensors in our ongoing cheap electronic modules from Asia (China).

Other versions

There are some existing designs which do this job, but most seem to be designed for 110-120VAC mains, as used in the US and some other countries. For example, Adafruit's PowerSwitch Tail performs a similar function to our design. You definitely wouldn't want to use these with 230VAC mains, as used in Australia, Europe and elsewhere. It would probably blow up and even if it didn't, it wouldn't be safe.

In the past, when we needed to control mains outlets using a microcontroller, we modified a 433MHz remote mains switch to do the job. The last time we did this was in the November 2015 issue – see *Programmable mains timer with remote switching*.

While simple and elegant, it's more expensive and more work, as you need to buy and modify the remote mains switch units.

So we have designed this unit, which is simple, cheap, reliable and able to switch just about any mains device, up to 10A rating. You could even connect several units to one micro to switch multiple devices.

How it works

The 230V *Opto-Isolated Mains Relay* uses a logic signal (eg, 3.3V, 5V or up to 10V) and switches a mains-rated relay on or off based on the state of that signal.

The optical isolation ensures that there is no chance that mains voltages could appear on the logic input and cause a shock hazard, or damage the driving circuitry. The 4N25 optocouplers we are using have an isolation rating of 5300V RMS.

It also has a logic signal output, which can be used by the driving circuitry to detect whether mains power is present and also allows the phase and frequency of the mains waveform to be sensed. This output uses the same type of optocoupler for safety.

An optocoupler consists of an LED (usually infrared) and phototransistor in a plastic package. The LED shines on the phototransistor junction through an insulating clear plastic section, so that the phototransistor conducts when the LED illuminates it. It behaves like a transistor with separate base-emitter and collector-emitter junctions. The gain (called the 'current transfer ratio', or CTR) is much lower than a standard transistor, so the collector current is generally of a similar magnitude to the LED drive current.

The CTR may be above or below 100%, depending on the particular optocoupler used (it can vary from sample to sample) and on the LED current; the CTR tends to peak at a few milliamps of LED drive current.

Now take a look at the circuit diagram shown in Fig.1. The control signal is applied to pin header CON2. When a sufficient voltage is applied, current flows through the 220Ω current-limiting resistor and through OPTO1's internal LED, which usually has a forward voltage of around 1V. So with 3.3V across CON2, around 10mA flows through it.

Assuming the 24V mains-derived power supply on the other side of the optocoupler is present, the phototransistor then acts as an emitter-follower, supplying voltage to the base of NPN transistor Q1 via a 10kΩ current-limiting resistor.

Q1 provides some gain so that sufficient current flows through the coil of RLY1 to latch its armature, connecting pins 1 and 2 of CON1, the mains terminal and connecting the live pin of the mains output socket to the mains input.

Diode D6 prevents OPTO1's internal LED from being reverse-biased if the voltage at CON2 is reversed, mainly to protect against damage from static electricity.

The outgoing live line also drives the LED in OPTO2 via a 100kΩ 1W mains-rated resistor and a simple half-wave rectifier consisting of diodes D7 and D8. So when the live voltage is above about 2V, D7 is forward-biased and current flows through the LED in OPTO2. As a result, current flows between the pins of CON3 during the positive half of the mains waveform, if RLY1 is latched on. CON3 can be connected between a microcontroller digital input pin and ground so that the micro's pin is pulled low in this case.

A pull-up of some sort is required on that pin, to ensure its state changes when OPTO2's output switches off; many micros have built-in pull-ups which can be enabled in software.

That micro can also measure the frequency of the pulses from CON3 to determine the mains frequency (this is usually pretty accurate, so it could be used as a reference) and the transitions are near the zero crossings.

There will be a slight phase shift due to the threshold being 2V rather than 0V, but this can be compensated for in software if accurate detection of zero crossings is necessary.

Mains power supply

RLY1 has a 24V DC coil because a higher coil voltage means a lower coil current for the same power, and we have limited current available to drive it with such a simple power supply.

Neutral is connected directly to the circuit ground and the supply current comes from the live conductor via a 470nF X2-class capacitor, which limits the average current and a 150Ω series resistor, which limits the inrush current. The resulting voltage is then rectified by a bridge rectifier comprising diodes D1-D4, and this is filtered to pulsating DC using a 100µF electrolytic capacitor.

A 470kΩ resistor across the X2 capacitor discharges it when the unit is unplugged, to minimise the risk of getting a (small) shock from the circuit.

If you consider what happens starting at a zero crossing, when the live voltage is rising, the right-hand side of the X2 capacitor rises to around 350V DC while the left-hand side is limited to around 25V, due to zener diodes ZD1 and ZD2, which are effectively across the output of the bridge rectifier. Thus, the X2 capacitor charges up to around 325V DC.

When the live voltage starts to drop again, current flow through this part of the circuit ceases, until the live voltage drops to around 300V DC. The left-hand side of the X2 capacitor will then be at about –25V and so current will flow through the other half of the bridge rectifier and the X2 capacitor will start to discharge.

It will have fully discharged when the live voltage is around –25V and then the capacitor will start to charge in the opposite direction and the whole process will repeat as live reaches –350V and then starts heading back towards 0V.

What do you use it for?

Have solar panels and a pool? You can use a light sensor and a real-time clock (RTC) module to switch the pool pump on during the day when solar power is available, or during off-peak hours if the weather is bad.

Own a different type of pump? Use a float switch to control a pump. Turn it on when a storage tank is full or off when empty. Some float switches are light duty and may fail when switching high currents. Using the Opto-isolated Mains Relay to buffer the signal from a float switch will save its contacts from burn out

Need to control a heating/cooling system? Add a temperature sensor (and/or an RTC) to build a custom thermostat.

Own an amplifier (or other appliance) without a power switch? Add a remote-controlled on/off switch to an amplifier, by merely adding an IR decoder to a microcontroller module, hooking up our isolated relay, writing a few lines of code and using a spare TV remote.

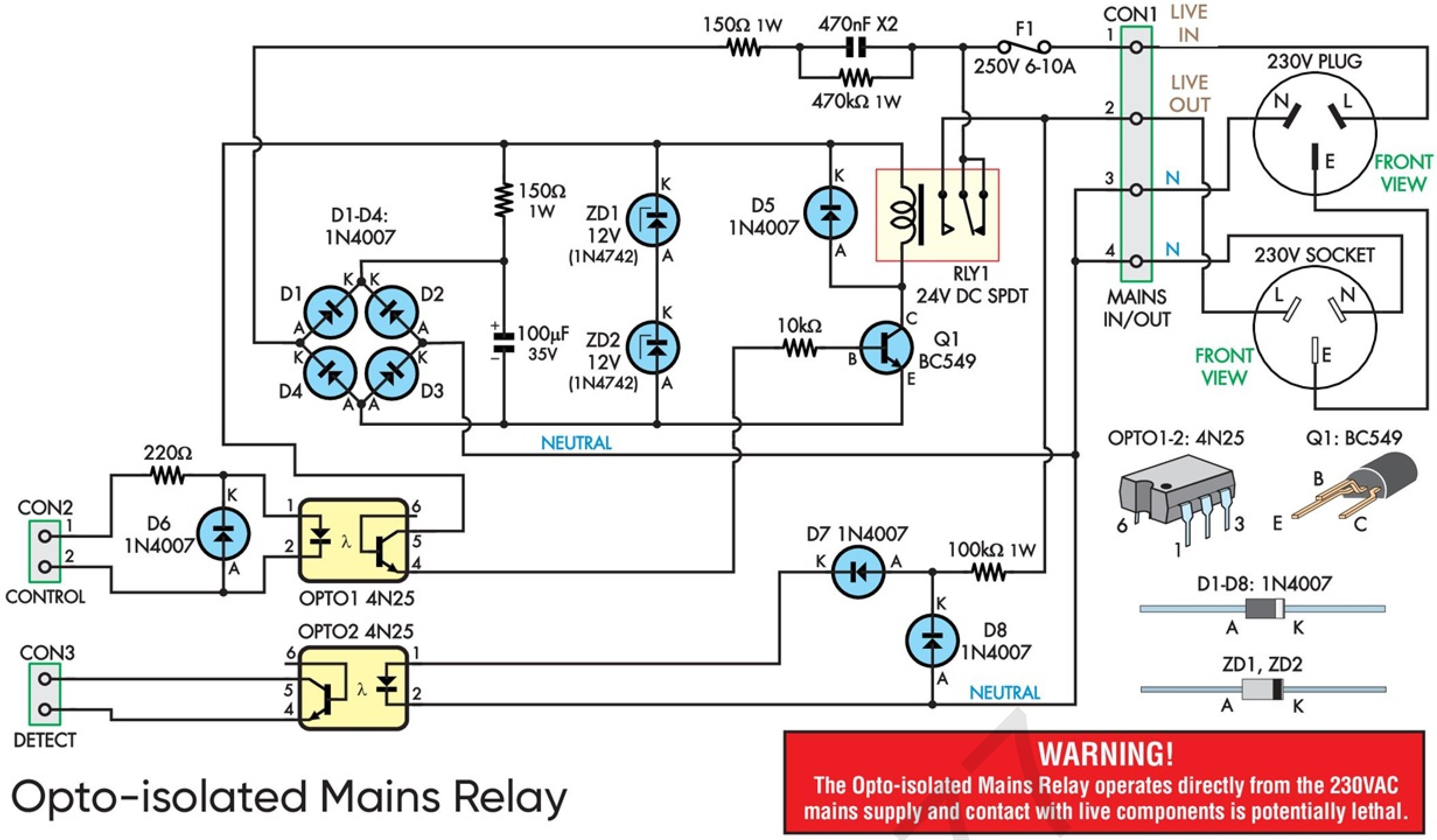
Unreliable Internet connection? Automatically reboot your router if your internet connection goes down, using a micro board with a Wi-Fi module.

What about a wireless power switch? Controlled by a micro or handheld remote control; just wire up a remote control receiver to its logic-level control input (the receiver needs a separate DC power supply).

Or you can implement a complex light switching arrangement, with multiple light switches controlling the same set of lights. Wire the switches to perform low-voltage signalling and then use this signal to drive the lights via the Opto-isolated Mains Relay. You could even use switches that are not mains-rated, incorporate remote control...

You don't even need a microcontroller to use the Opto-isolated Mains Relay. Any logic signal, from 2.7V up to about 10V can be used to activate the relay. This could come from an op amp output, logic gate, relay, switch, battery, plugpack or any other source of switched DC.

WARNING: this project involves mains voltages which can be dangerous if not handled correctly. Follow the instructions in this article carefully.



Opto-isolated Mains Relay

WARNING!
The Opto-isolated Mains Relay operates directly from the 230VAC mains supply and contact with live components is potentially lethal.

Fig.1: the complete circuit and wiring diagram for the *Opto-isolated Mains Relay*. The control signal at CON2 drives the LED in OPTO1, which switches NPN transistor Q1 to activate relay RLY1. The incoming mains live voltage is applied to a 470nF X2 capacitor, then rectified by diodes D1-D4 and filtered by a 100µF capacitor to provide around 25V DC for the relay coil. OPTO2 allows mains phase sensing and indicates when the load has mains power.

This process repeats continually, maintaining the charge across the 100µF capacitor at around 30V, drawing just a few milliamps from the mains.

Voltage regulation

When there is no signal at CON2 and RLY1 is off, the two zener diodes keep the positive end of RLY1's coil at around 24V; this is more than enough voltage to allow its armature to latch.

When Q1 does pull in, it diverts some but not all of the current that was flowing through the zener diodes to the relay's coil instead.

At 50Hz, the 470nF capacitor has an impedance (reactance) of about 6.8kΩ, limiting the current drawn from the mains (230VAC) to around 33mA. The 24V relay draws around 22mA at 24V, so the current through the zener diodes drops from around 33mA to around 11mA.

This assumes the mains is at the nominal 230V. These numbers change if the supply voltage changes, and so the extra current means the relay will work reliably even with mains voltages slightly below 230V.

ZD1 and ZD2 also limit the voltage across the 100µF filter capacitor to a safe level. We have used two 12V zeners rather than one 24V zener because the total dissipation with RLY1 off is not much below 1W and could be higher if the mains voltage is elevated.

The second 150Ω series resistor, between the 100µF capacitor and relay coil, helps to prop up the coil voltage for the first few milliseconds after Q1 switches on, ensuring correct latching.

This works because the 100µF capacitor can charge to a slightly higher voltage initially, due to the voltage across this added resistor.

Diode D5 protects Q1 from voltage spikes caused by back-EMF when RLY1 switches off; while fuse F1 blows if there is a fault on the mains side of this circuit, or if the load goes short-circuit, preventing any further circuit damage.

As noted, RLY1 requires around 22mA to operate. Q1's h_{FE} is typically over 400, meaning a base current of 55µA is needed to activate the relay. Assuming that OPTO1's CTR is at least 20%, that means the driving circuitry needs to be able to supply around 0.3mA at a minimum voltage of about 2.2V, to switch on the relay.

Construction

As with any circuit involving mains voltages, it is imperative that the case and mechanical construction are completed correctly to ensure the safety of the completed circuit.

Attention to detail when building the PCB is critical too, as a single reversed diode could destroy other components in the circuit before the fuse has a chance to blow.

The *Opto-Isolated Mains Relay* is built on a PCB coded 10107181 which measures 99 x 60mm and is available from the *PE PCB Service* – see page 78. The PCB is designed to clip into the internal side rails of a UB3 Jiffy box, leaving just enough room at the end of the box to fit two cable glands, which are used to secure the mains cables. Use the PCB overlay diagram, Fig.2, as a guide during assembly.

The first step is to fit the low-profile passive devices, starting with the resistors. It's best to check the values with a multimeter before soldering them in place where shown in Fig.2.

Fit diodes D1-D8 next. Take care to ensure you insert them with the cathode stripe in the orientation shown in the overlay diagram. Note that D1 and D2 face the opposite direction to D3 and D4. Then mount the two identical zener diodes. Again, ensure that the cathode band is oriented correctly.

Q1 is the only transistor, and it should be oriented as shown in Fig.2. You may need to bend and adjust the legs to fit the holes on the PCB (eg, using smaller pliers).

The two optocouplers, OPTO1 and OPTO2, should be soldered next. Note that the notches on the packages both point in towards the centre of the board.

The PCB has been slotted to reduce the chance of leakage between the two halves of the board (ie, increase the creepage distance), so we have added a dot adjacent to the number one pin in each case. Align this with the dot on the optocoupler packages.

If you are going to install header terminals for CON2 and CON3, now is a good time to do so. You could instead solder wires directly to these pads later. If you are installing the extension pieces, fit the header terminals with the long pins down.

Next, fit the fuse holder clips to the board. Make sure the retention tabs are facing towards the outside or else the fuse will not fit. You can temporarily install the fuse to make sure the holder clips are placed correctly, but be careful if you solder the clips with the fuse in place, as the heat could damage the fuse (eg, you could accidentally desolder the end caps). Remove the fuse after soldering.

Install the electrolytic capacitor next. It is polarised, so it must be fitted the right way around. The stripe on the capacitor body indicates the negative lead while the positive lead is longer.

The positive lead should go into the pad marked with a '+' sign. We have specified a 50V capacitor, but a 35V or 63V-rated capacitor would work fine.

Solder the barrier terminal in place now. If you're using the Jaycar version, which we prefer, you should attach it to the board using two machine screws, washers and nuts before soldering the pins.

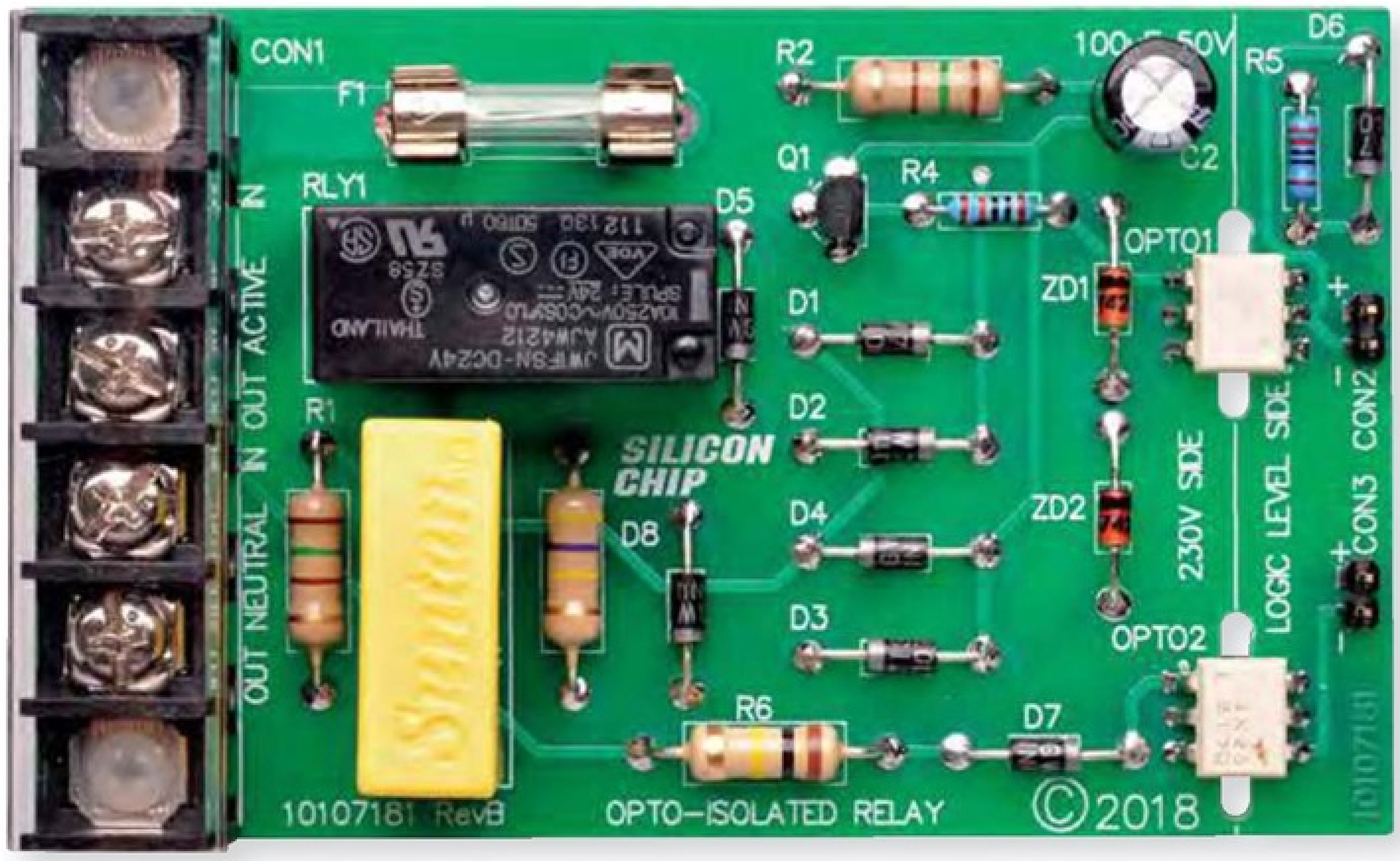
These screws prevent any stress on the solder joints. But the cable glands we're going to use to clamp the mains cords should also prevent stress, so the Altronics version without the moutning screw holes should be OK too.

The terminals on the barrier terminal are quite large, so you may need a bit of extra solder and heat to ensure a good mechanical and electrical connection.

Finally, fit the X2 capacitor and relay. Both should be pushed down fully onto the PCB before soldering. The capacitor is not polarised while the relay can only go in one way.

Putting it in the box

You must mount the PCB in the Jiffy box to provide sufficient insulation to make it safe. Start by drilling two holes at the end of the UB3 Jiffy box to suit the cable glands. The specified glands require 16mm holes. If you are using a different gland, you may need a different hole size. Use the cutting template in Fig.3 as a guide – it is available for download from the October 2019 page of the *PE* website.



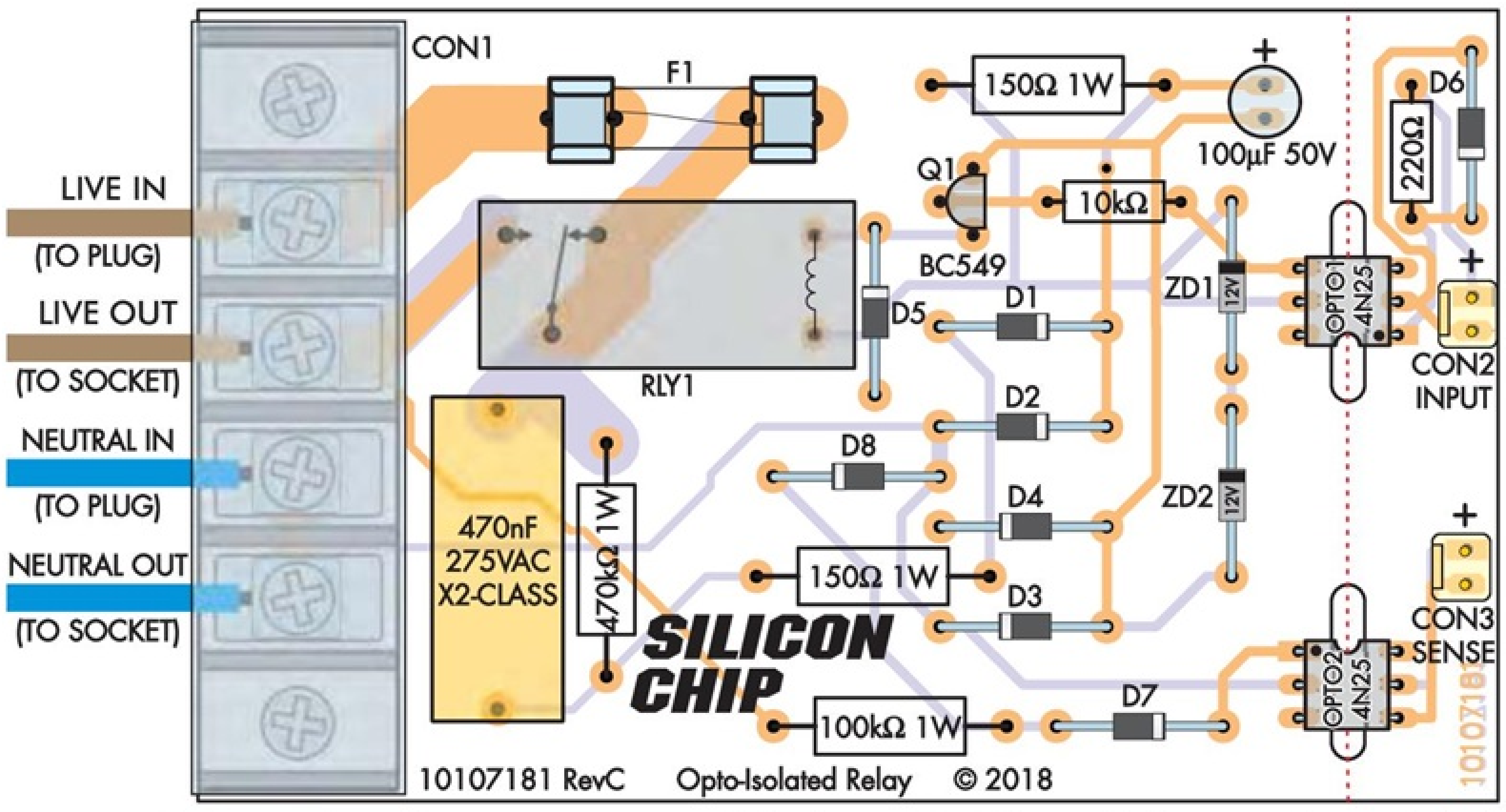


Fig.2: use this overlay diagram to assemble the PCB. The safe, low-voltage side is at the right, while the rest of the board is connected directly to the mains. During construction, take care with the orientation of the diodes and the electrolytic capacitor. The PCB should be sealed in its box before plugging it in. The layout has changed slightly since the prototype was built, to increase track clearances.

The two slots on one end of the box are designed to provide access to CON2 and CON3. You could simply solder some light-duty figure-8 wires to those pads, or use twin leads with DuPont headers on the end to plug into the pin headers.

We've also prepared two slim PCBs, which are supplied along with the main PCB and these can be soldered to the board in place of CON2 and CON3. They then pass through slots in the case and have mounting pads for small terminal blocks, which sit just outside the plastic case and make it easy for you to attach wires for connection to your control module.

Regardless of which approach you take, we suggest you make the slots anyway, since you need some way to get the control signals into the case. They can be made by drilling a few small holes in a row (eg, using a 2mm drill bit) and then joining and shaping them with a needle file.

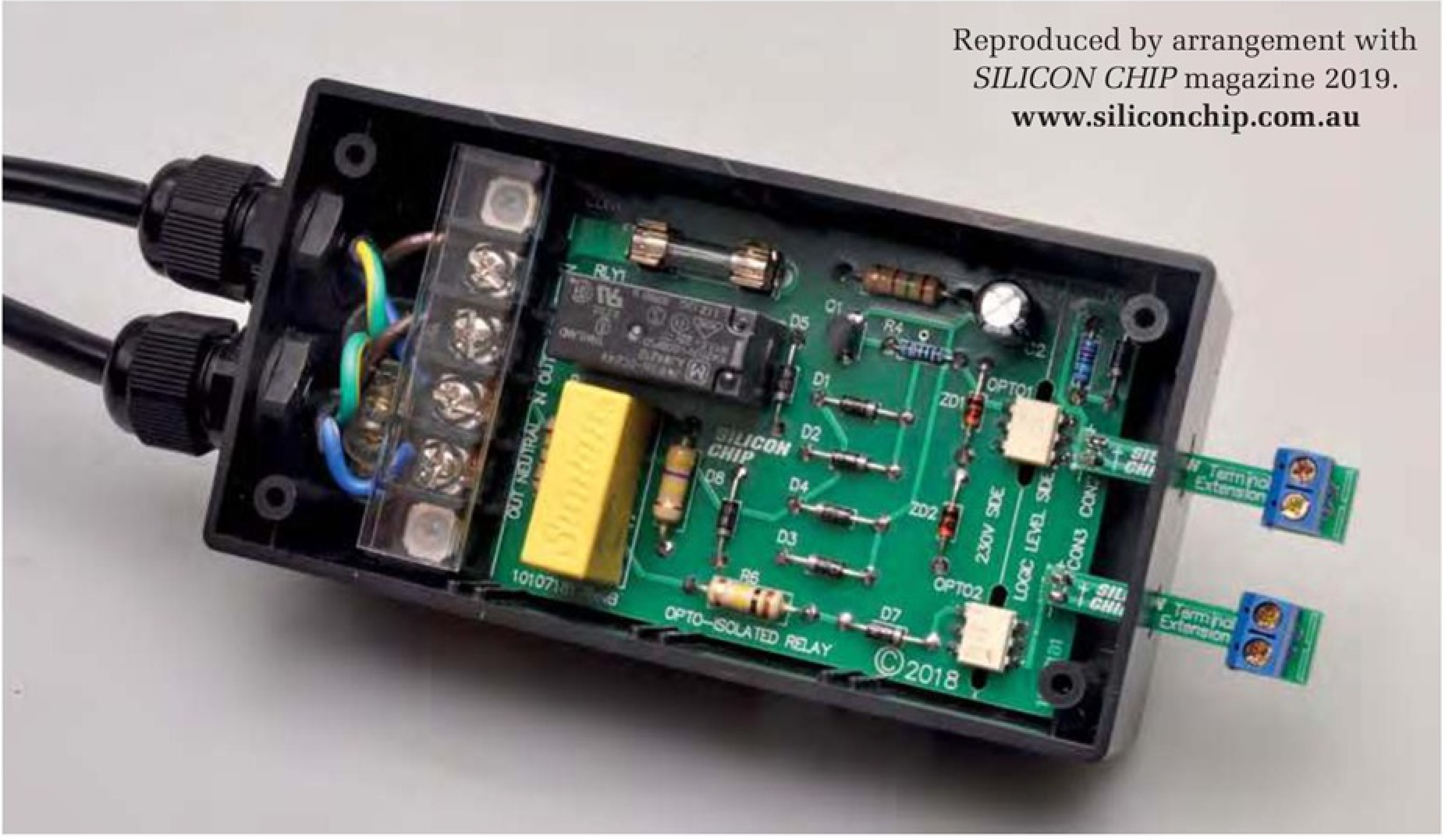
You could drill a larger hole, but that would make it easier for dust and debris to get inside the box. You certainly shouldn't make these holes any larger than necessary to prevent wires from accidentally poking inside the case, which could (in an admittedly unlikely scenario) make contact with live portions of the board.

A step drill is handy for drilling the larger holes for the cable glands, but if you don't have one, you can use a tapered reamer instead. Fit the cable glands to the enclosure and make sure the nuts are done up tight. We found that the lips on the mounting nuts overlapped slightly, so we trimmed them with a sharp pair of sidecutters.

Preparing the mains cable

It's up to you where to cut the mains cable to form the two leads. You could cut it in the middle to get two equal-length cords, or you could make the plug or socket end longer, depending on your application. Make sure there is at least 30cm of cable left at each end after cutting it.

Once you have cut the cable, there are exposed ends that present an electrocution risk if the plug is connected to a socket. Take great care to ensure that the plug cannot be plugged into a socket while you are working on it (or if you leave it unattended). It helps to



Reproduced by arrangement with
SILICON CHIP magazine 2019.
www.siliconchip.com.au

SILICON CHIP
Terminal Extension
+
–

Left: the *Opto-isolated Mains Relay* mounted inside a UB3 Jiffy box with the mains power plug and power socket wired up. Note the terminal extension boards, as shown above, are wired positive (+) to positive. These are optional attachments to make connecting your control module easier.

plug the plug end into the socket end until you have finished wiring it up.

Feed the cut ends of the cable through the glands. The plug end should go through the gland closest to the fuse. Make sure to thread the domed nut onto the cable first, if you had to remove it.

Strip back the outer insulation by 25mm on both ends, then strip the insulation back by 5mm on the live (brown) and neutral (blue) wires. The earth wires (green and yellow stripes) should be stripped back about 15mm.

Remove the clear barrier from the terminal barrier and attach the wires as shown in Fig.2. The top screw terminal takes the incoming live (brown) from the plug lead. The next screw terminal is for the outgoing live wire to the socket, also brown. Ensure both of these are firmly screwed down.

The bottom two screws are for the two neutral wires (blue) and they are connected together on the PCB. While it will work regardless of which wire goes to which screw, it is neater to connect the incoming (plug) neutral wire to the third terminal and the outgoing (socket) neutral wire to the bottom screw terminal.

The two earth leads should now be joined using the BP-style double screw connector. Twist the wires together and then insert them into the connector, making sure that they both reach all the way to the end, then do up both screws tight and check that they have both clamped the wires.

Now check your work to ensure there are no exposed copper strands from any of the wires floating around or touching the wrong terminals and then replace the transparent barrier strip over the barrier terminal. Check also that none of the wires can move around in their respective screw terminals.

To test the unit, insert a 100mA (or similar current) fuse in the holder and slot the PCB into the grooves in the enclosure, then tidy up the wires using cable ties. You can tuck the BP-style screw connector under the gland entry inside the box. Check that around 5mm of the outer mains cable insulation is visible inside the box before firmly tightening the glands. This ensures that the glands grip the cables securely.

Testing

The first tests (with the low-value fuse in place) are to verify there are no problems with the PCB construction. Don't connect anything to the mains socket yet. Place the unit somewhere stable and during testing, stay well away from it – don't touch anything inside the box. Plug the unit into a switched-off mains socket and then switch it on.

If the fuse blows or the relay activates (you will hear it click), you may have mis-wired something. Turn off the power point and **unplug the plug**.

You can test for residual charge by carefully connecting a multimeter on a high DC volts range across the live and neutral pins of the mains plug. If there is voltage present after a few seconds, your bleed resistor may not have been fitted correctly.

If all is well, nothing obvious should happen. Turn off the power point, unplug the unit and connect a 3.3-10V DC voltage source to CON2 with the

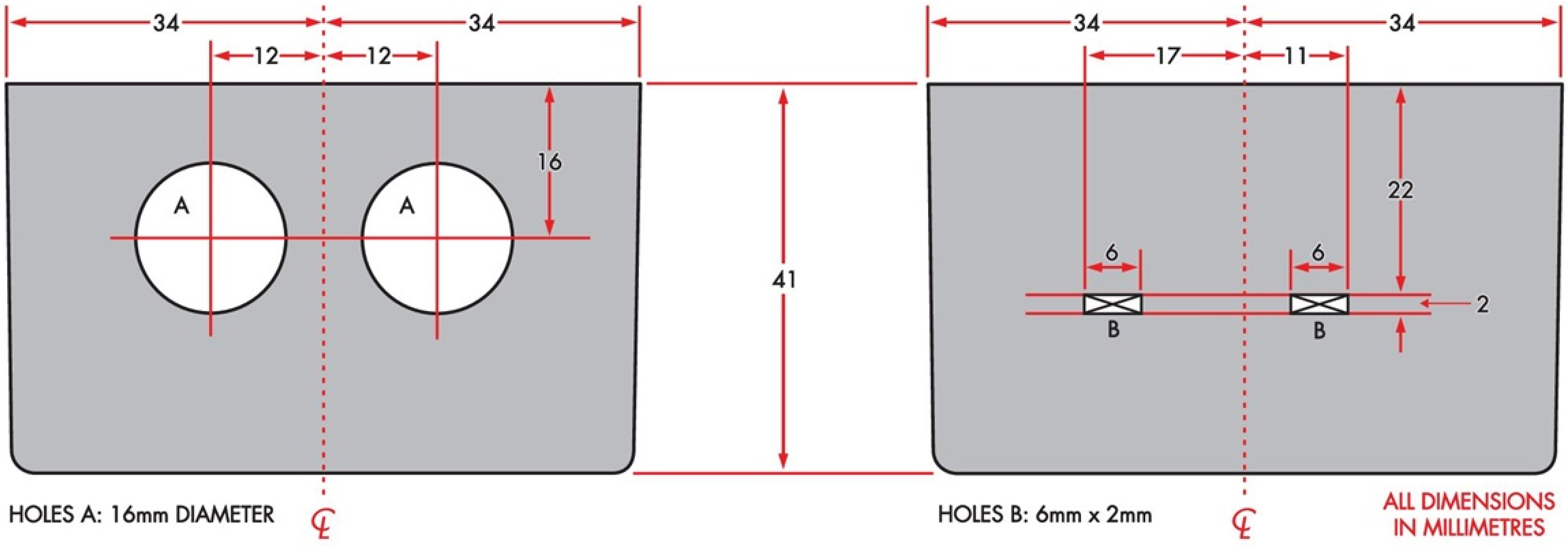


Fig.3: drilling and cutting diagram for the UB3 plastic Jiffy box, reproduced same size (available for download from the October 2019 page of the *PE* website). The two 16mm holes are for the cable glands that clamp the mains cords, while the slots are for either figure-8 wires or extension PCBs to give access to the isolated control and feedback signals. The slots can be made by drilling a series of small holes which are then joined and shaped using a needle file.

indicated polarity. Turn on the power again and check that the relay clicks as the armature pulls in. That shows that the circuit is working.

Turn off the power, unplug the lead and replace the fuse with the final value. For example, if you are using the Jaycar 5A relay, the fuse rating should be no higher than 5A. If you are using the 16A relay from Altronics, use a 10A fuse, as the mains leads cannot safely carry a higher current.

Testing the mains presence/phase output

The easiest way to test that the CON3 output is working is to connect a high-brightness LED with its cathode to pin 1 of CON3, then connect the anode to pin 1 of CON2 (the positive control signal input) and wire pin 2 of CON3 to pin 2 of CON2. You still need to apply the DC voltage to CON2 since the CON3 output is only active when the relay is latched.

If you plug the unit back into mains and switch it on, you should find that the LED lights when the relay is engaged and switches off when you cut mains power. It will actually be flashing at 50Hz with a ~50% duty cycle, but this may not be obvious to the naked eye. The LED current will be limited to one or two milliamps due to the limited CTR of OPTO2.

Finishing it off

Before you put the lid on, if you haven't already done so, make the control connections to CON2 and CON3. If using our small extension boards, fit the terminal blocks on the wider end, then feed the boards through the slots in the case (lining up + with + and – with –) and place the holes in the extension PCBs over the header pins. You can then solder them in place.

Note that once this has been done, they need to be desoldered to remove the PCB, so it is important that everything is working and the lid fits correctly before doing this.

If soldering wires to the pads for CON2 and CON3, pull them tight and then glue them into the holes in the box with silicone sealant. This ensures that if the solder joints fail, the wires cannot come in contact with the high-voltage section of the PCB.

Now screw the lid on, to ensure that no live parts are exposed. Also, unwind the cable gland nuts and add a few drops of super glue to the threads, then do them up tight again. This stops anyone from undoing them while the device is plugged in. The *230V Opto-isolated Mains Relay* is now complete and can be used for your intended purpose.

Parts List – Opto-Isolated Mains Relay

1 double-sided PCB, coded 10107181, 99mm x 60mm
2 double-sided PCBs, coded 10107182, 38mm x 10.5mm (optional)
[Both PCBs are available from the *PE PCB Service* – see page 78.]
1 230V 10A extension cord (or mains plug and socket with leads)
2 cable glands to suit mains cord
  [eg, Jaycar HP0724 or Altronics H4312A/H4313A]
1 UB3 Jiffy box [Altronics H0153/H0203 or Jaycar HB6013/HB6023]
1 4-way PC-mount terminal barrier (CON1) [Jaycar HM3162 or Altronics P2103]
1 BP-style double screw connector [Jaycar HM3192 or Altronics P2125A]
1 250V-rated 24V DC coil relay (RLY1)
  [Altronics S4199 (16A, recommended) or Jaycar SY4051 (5A)]
2 M205 PCB-mount fuse clips (F1) [Jaycar SZ2018, Altronics S5983]
1 M205 slow-blow fuse to suit relay contact rating, no more than 10A
  [Altronics S5662, Jaycar SF2178]
1 M205 100mA or similarly rated fuse (for initial testing only)
2 2-way headers, 2.54mm pitch (CON2,CON3)
2 M3 x 20mm nylon machine screws
2 M3 nylon hex nuts
2 M3 nylon flat washers

Semiconductors
8 1N4007 1A 1000V diodes (D1-D8)
2 12V 1W zener diodes, eg, 1N4742 (ZD1,ZD2)
2 4N25 optocouplers (OPTO1,OPTO2)
1 BC549 100mA NPN transistor (Q1)

Capacitors
1 470nF 275VAC X2-class MKT/MKP
1 100µF 50V RB electrolytic

Resistors (all 1W, 5% unless otherwise stated)
2 150Ω 1 470kΩ 1 100kΩ 1 10kΩ 0.25W 1% 1 220Ω 0.25W 1%

If you are planning to use the output/phase sense signal from CON3 with an Arduino, you can enable an internal pull-up current on the digital input pin using a command like this, within your setup() function:

```
pinMode(3,INPUT_PULLUP);
```

In this example, connect digital pin 3 to the + terminal of CON3 and GND to the – terminal of CON3. When the mains is off, pin 3 will read high (1), while you would get a low reading (0) during the positive-going half of the mains cycle when the relay is on. You could use a pin change interrupt or counter function to detect the pin toggling if you simply need to know whether the load is powered.

In the event of a blackout or if, for some reason, the relay fails to close, that pin will remain high. You can detect that condition and flag an error (eg, by sounding a buzzer).

Mains phase detection is possible using this signal, but it's a little bit complicated due to the phase shift – you need to use a timer to measure the positive and negative times, calculate the delay between the zero crossing and the pin going low, then use another timer (or possibly the same one) to compensate. That's a bit too much detail to get into here.

You don't necessarily need to use the output sensing function, though. You can leave CON3 disconnected if you do not need that feature.

The finished *Opto-isolated Mains Relay*. The two small extension boards make it easy to connect the low voltage isolated terminals to a suitable controlling module.

UK constructors will need to use a UK mains plug/socket.

PCBWay

Win $50 worth of PCBWay prototyping!

PCBWay is offering Practical Electronics readers the chance to win one of six $50 credits to use with their world-leading PCB prototyping service.

Based in Shenzhen, China, PCBWay not only manufactures and assembles PCBs, but also offers its clients the most user-friendly, web-based interactive design system across the industry.

Online Quote — Upload PCB File — Order Review — Payment — Fabrication — Delivery — Confirm Received

PCBWay's online web system lets you instantly get a quote for your PCB or PCBA order, and then track the online the status of its fabrication – all via your personal PCBWay account panel. After your PCBs have been dispatched, their shipping status can be checked on your order management page.

Our online facilities help keep everything under control, saving you time and energy all too often wasted in a frustrating loop of emails and phone calls.

How to enter

For your chance to win a $50 free trial of PCBWay's prototyping service, just create your own PCBWay account at:

http://bit.ly/pe-oct19-pcbway

Closing date: The closing date for this offer is 1 October 2019

CRICKLEWOOD ELECTRONICS
Established 1981

Your best bet since MAPLIN
Chock-a-Block with Stock

Visit: www.cricklewoodelectronics.com

Or phone our friendly knowledgeable staff on 020 8452 0161

Components • Audio • Video • Connectors • Cables
Arduino • Test Equipment etc, etc

Visit our Shop, Call or Buy online at:
www.cricklewoodelectronics.com

020 8452 0161

Visit our shop at:
40-42 Cricklewood Broadway
London NW2 3ET

ESR Electronic Components Ltd

All of our stock is RoHS compliant and CE approved. Visit our well stocked shop for all of your requirements or order on-line.

We can help and advise with your enquiry, from design to construction.

3D Printing • Cable • CCTV • Connectors • Components • Enclosures • Fans • Fuses • Hardware • Lamps • LED's • Leads • Loudspeakers • Panel Meters • PCB Production • Power Supplies • Relays • Resistors • Semiconductors • Soldering Irons • Switches • Test Equipment • Transformers and so much more…

Monday to Friday 08:30 - 17.00, Saturday 08:30 - 15:30

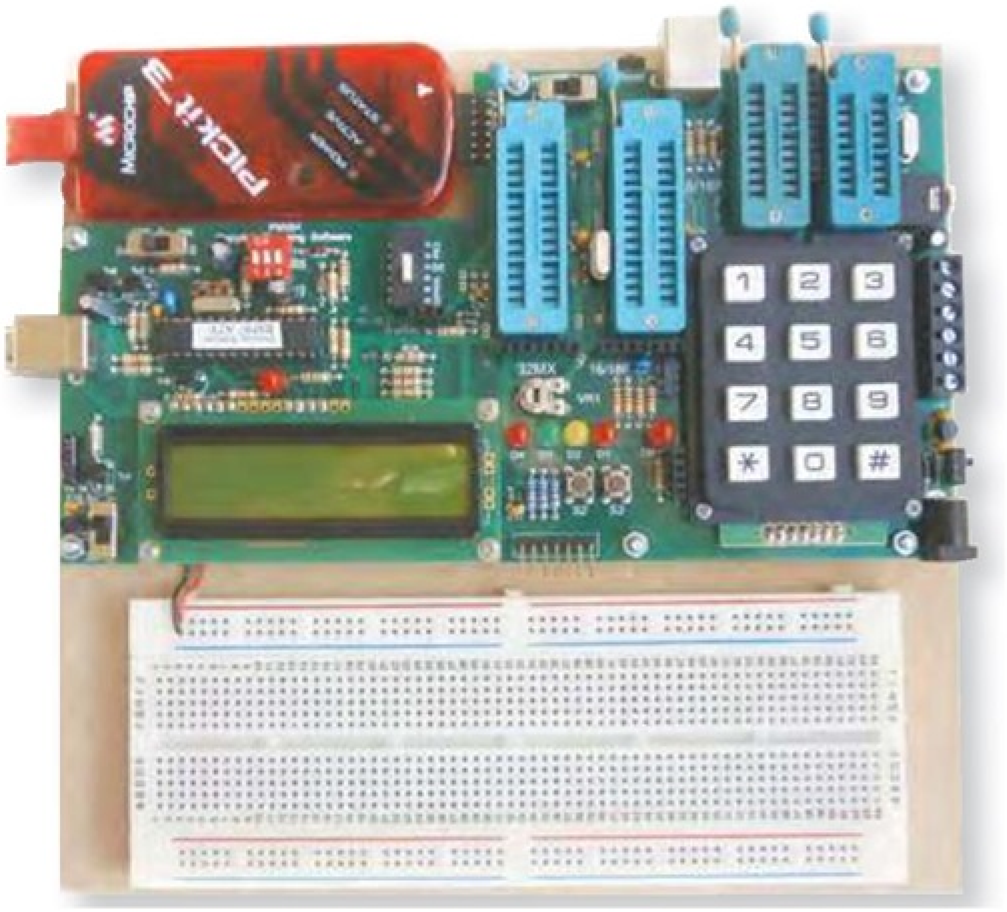
ARDUINO
cebek
micro:bit
velleman

Station Road
Cullercoats
North Shields
Tyne & Wear
NE30 4PQ

Tel: 0191 2514363 sales@esr.co.uk www.esr.co.uk

P955H PIC Training Circuit

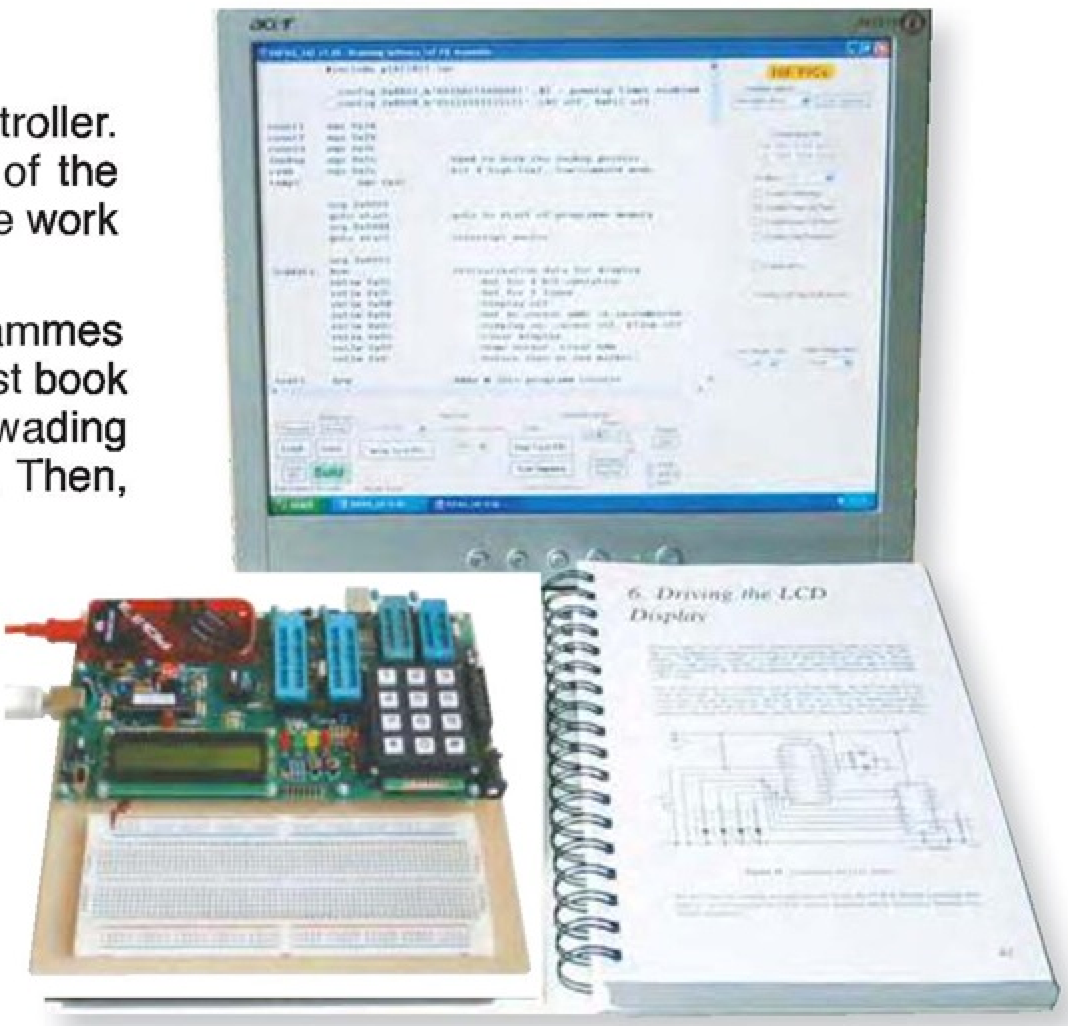
by Peter Brunning



When you are first learning about PICs, whether you are a complete beginner or an experienced programmer, you need an uncomplicated system which allows you to learn without getting bogged down in system procedure. That is why we created the P955H PIC training circuit and our own PIC assembler. In the first book we learn about PIC programming using the Brunning Software PIC assembler BSPWA, but in chapter 3M there is an introduction to the Microchip assembler MPASM X. All our assembler text will run in both systems, so from there on, if you wish you can use MPASM X. Likewise, we start by using the on-board PIC programmer to write the code into the PIC, but if you prefer, plug in a PICkit 3 and use that. The P955H training circuit has the flexibility to be what you need as your learning process advances.

The P955H training circuit has been designed to work with both 32-bit and 8-bit PICs. The idea is to start learning about PICs using assembler with 8-bit PICs. Then learn C with 8-bit PICs, study serial communications using 8-bit PICs, and finally study C programming using 32-bit PICs. It is a simple approach to a subject that has no limit to its ultimate complexity.

The Brunning Software P955H PIC Training Course



We start by learning to use a relatively simple 8-bit PIC microcontroller. We make our connections directly to the input and output pins of the chip and have full control over the internal facilities of the chip. We work at the grassroots level.

The first book teaches absolute beginners to write PIC programmes using assembler, which is the natural language of the PIC. The first book starts by assuming you know nothing about PICs, but instead of wading into the theory we jump straight in with four easy experiments. Then, having gained some experience, we study the basic principles of PIC programming, learn about the 8-bit timer, how to drive the alphanumeric liquid crystal display, create a real-time clock, and experiment with the watchdog timer, sleep mode, beeps and music. Then there are two projects to work through. In the space of 24 experiments, two project and 56 exercises we work through from absolute beginner to experienced engineer level using the latest 8-bit PICs (16F and 18F).

The second book introduces the C programming language for 8-bit PICs in very simple terms. The third book, *Experimenting with Serial Communications*, teaches Visual C# programming for the PC so that we can create PC programmes to control PIC circuits.

In the fourth book, we learn to programme 32-bit MX PICs using fundamental C instructions. Flash the LEDs, study the 16-bit and 32-bit timers, write text to the LCD, and enter numbers using the keypad. This is all quite straightforward as most of the code is the same as already used with the 8-bit PICs. Then life gets more complex as we delve into serial communications with the final task being to create an audio oscilloscope with advanced triggering and adjustable scan rate.

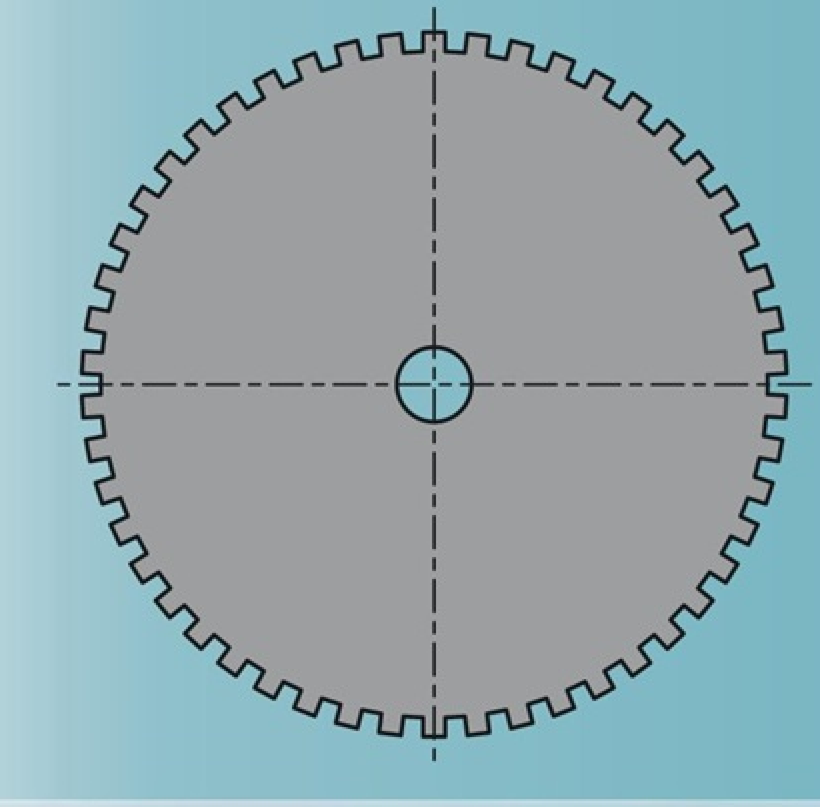
The complete P955H training course is £259, which includes the P955H training circuit, four books (240 × 170mm, 1200 pages total), six PIC microcontrollers, PIC assembler and programme text on CD, two USB-to-PC leads, a pack of components, and carriage to a UK address. (To programme 32-bit PICs you will need to plug in a PICkit 3, which you can buy from Microchip for £38.)

Prices start from £175 for the P955H training circuit with Books 1 and 2 (240 × 170mm, 624 pages total), two PIC microcontrollers, PIC assembler and programme text on CD, USB-to-PC lead, and carriage to UK address. (PICkit 3 not needed for this option). You can buy Books 3 and 4, USB PIC, 32-bit PIC and the components kit as required later. See the Brunning website for details: **www.brunningsoftware.co.uk**

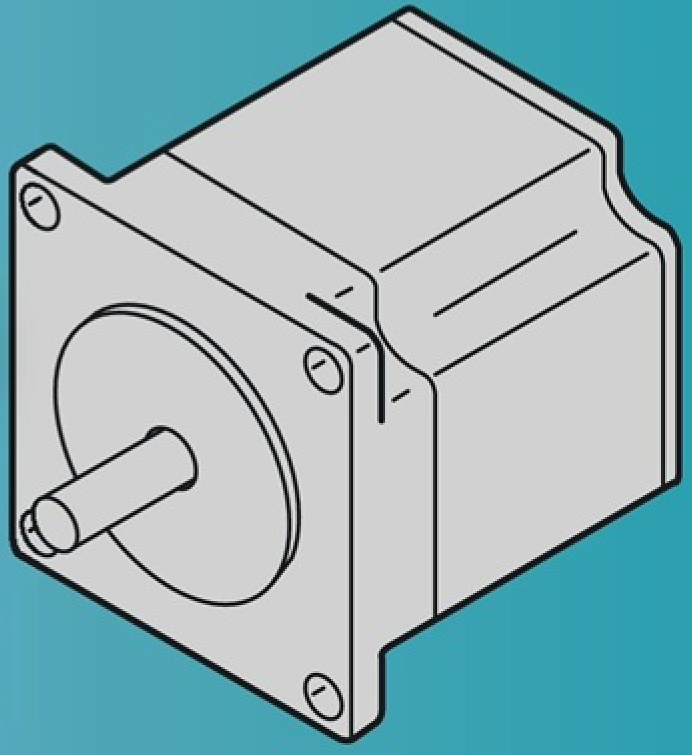
Mail order address:

Brunning Software

138 The Street, Little Clacton, Clacton-on-sea,
Essex, CO16 9LS. Tel 01255 862308



Using Stepper Motors



Part 1 | by Paul Cooper | technobotsonline.com

Introduction to stepper motor technology

For 18 years, I have been answering customer's technical questions at: **technobotsonline.com** Technobots supplies electronic and mechanical components to hobbyists, trade and education. A recurring topic from customers has been the use of stepper motors and how to control them. Despite stepper motors being around since the 1950s and in common use since the 70s, they remain a bit of a mystery to many hobbyists. In this series of articles, I will cover both their theory and how to use them, culminating in a full 3-axis CNC controller. On the way, various controller designs will be covered; so whether you simply want a slow rotating radar dish on your model ship, or perhaps a motorised camera slider, these articles will show you the way. By the end of this first article, you should have an insight into the main types of stepper motor available.

What is a stepper motor?

The everyday *brushed* DC motor requires little introduction; it has powered everything from small toys to powerful machine tools for over a century. Connect it to a DC supply and it will spin at many thousands of RPM. Reverse the polarity and it will spin in the opposite direction.

DC motors, therefore, are one very effective way to supply rotational *speed* control; but what if you didn't want to just control rotational speed, but rotational *position* too? This is where stepper motors are used. However, stepper motors are not as straight forward as DC motors; a simple pot won't do the job – they do require an electronic controller to manage the energising of the motor coils.

A stepper motor is a *brushless* DC motor that moves in discrete steps in either direction and can be made to stop and hold at any of these step positions. Immediately, the advantages of stepper motors over DC brushed motors should be apparent for applications where positional control or a slow rotational speed is required without the need of a positional feedback sensor.

Stepper motor applications

Stepper motors can range in size from milliwatts to hundreds if not thousands of watts, and are found in numerous commercial products including:

- Wrist watches
- 3D printers
- Photocopiers and printers
- CNC machine tools
- Consumer electronics such as camera focus and zoom control
- Digital pumps for accurate metering of liquids
- Robotics
- Throughout the automation industry
- Legacy computer accessories – floppy disk drives

Unlike brushed DC motors, stepper motors draw most current when they are *not* rotating due to their holding current. Some stepper drivers will reduce the holding current to a motor until it is needed to rotate again. The holding current and lack of fan-assisted forced-air cooling means that steppers can run rather hot, especially when they are stationary.

Types of stepper motors

This series of articles is not about the internal operation of stepper motors – fascinating though that is – but how to use them. If you want a good introduction to how they work, then the YouTube video mentioned below in the *Hybrid stepper motors* section is an excellent place to start. That said, we can't treat them purely as black boxes, so here is a quick overview of the three main types of stepper motor:

- Permanent magnet
- Variable reluctance
- Hybrid

The key differences are found in the rotor (the part of the motor that rotates) and the stator (the stationary windings wrapped around the rotor). You will see the term 'soft iron' used. Soft iron is annealed, and unlike magnetically 'hard' iron, it has low coercivity, which means it does not remain magnetised when the magnetic field from the stator coils is removed. This is vital in applications where the magnetic field is required to be repeatedly switched on and off.

Permanent magnet stepper motor

These motors have an axially magnetised cylindrical rotor with alternating north and south poles parallel to the rotor shaft. In its simplest form, the stator usually consist of two windings, each wound on diametrically opposite poles in series to form a two-phase winding enclosed in a metal can. In practice, more poles are

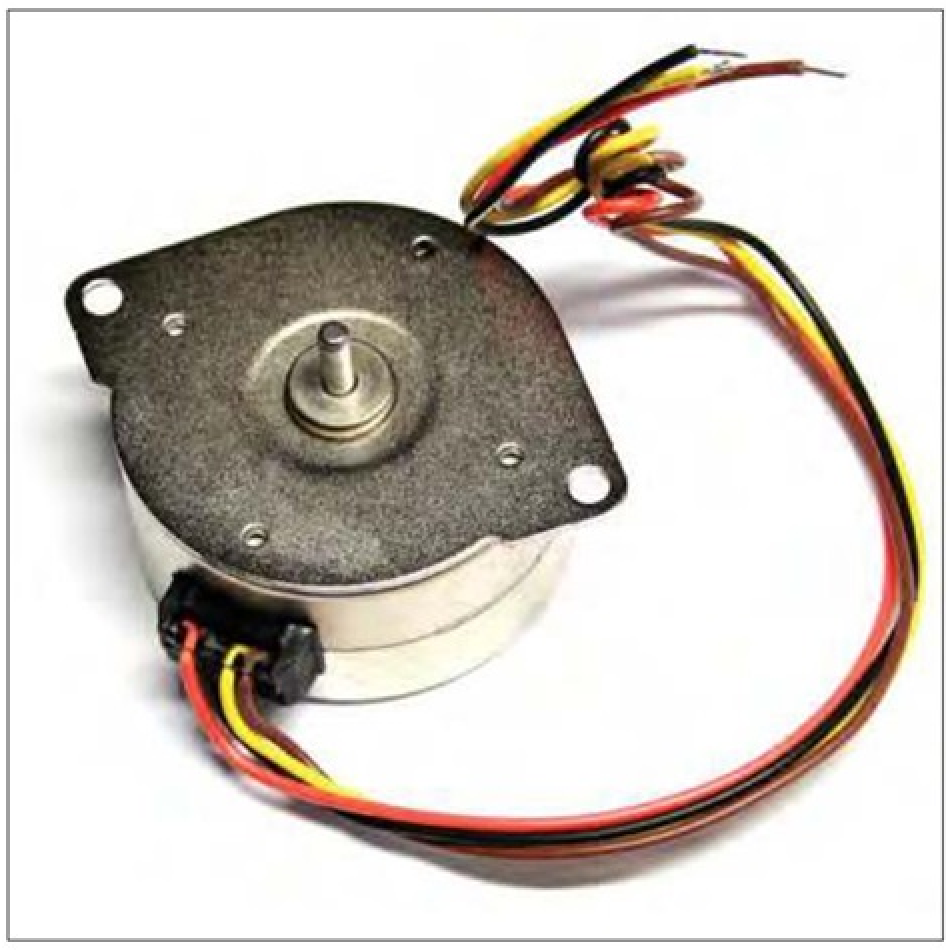


Fig.1. A 48-step bipolar permanent-magnet stepper motor with two stacked 'cans'.

FD&19MX77

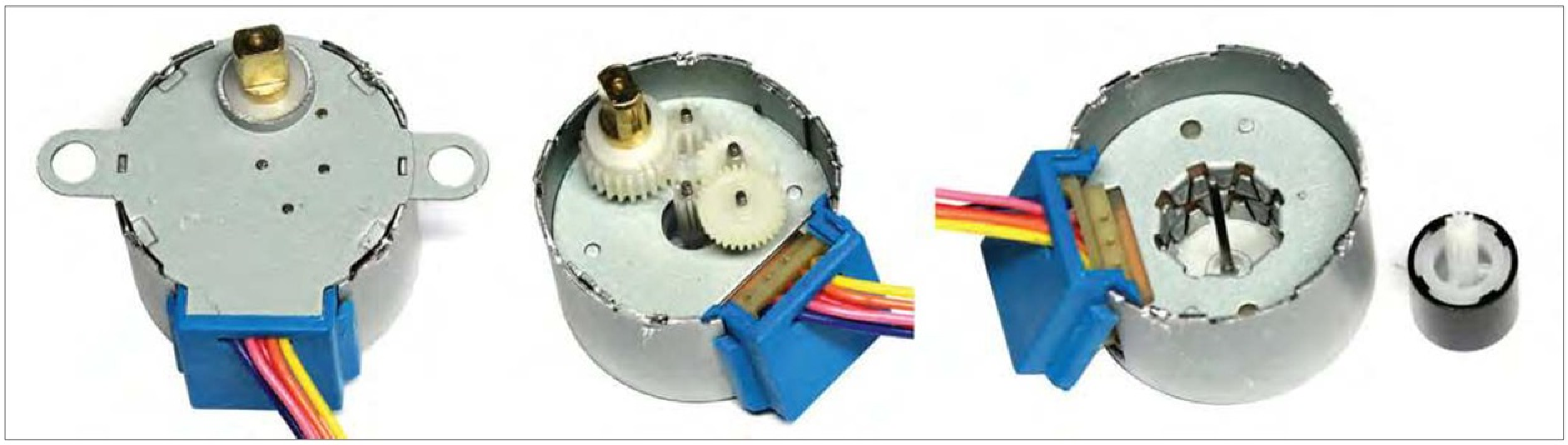


Fig.2. Internal view of a 28BYJ-48 permanent-magnet stepper motor with integral gearbox. Note the 'fingers' around the inside of the stator giving 32 steps per revolution. The number of step at the output shaft is approximately* 32 x gearbox ratio 64 = 2048 in full-step mode. (*note that the gearbox ratio is not exactly 64:1).

needed to create more steps per revolution which is where 'can stacked' motors (Fig.1) come in. To create more poles but keep to the two windings is achieved through the use of 'fingers' or 'teeth' around the inside of each coil stacked. Typically such steppers are available with 12, 20, 24, 48 and 100 steps per revolution, sometimes higher, but 24 and 48 are common. The low step count can be offset with models having integral gearboxes, the 28BYJ-48 stepper motor (Fig.2) with its approximate 64:1 ratio gearbox can be obtained with a driver for under £2 with free carriage on eBay! Not bad for a stepper motor that can achieve over 2,000 steps per revolution in what is called 'full-step mode'. (Step modes will be described in a later article.)

The key advantages of permanent magnet stepper motors are high suitability for low-speed applications with moderate torque requirements and they tend to be less expensive than hybrid motors.

Variable-reluctance stepper motors

These motors do not have a permanent magnet in the rotor, but one made of soft iron / slotted steel laminations. The soft iron rotor is multi-toothed and offers good angular resolution and rotational speed but lacks torque compared to the other types. One significant difference of this type of motor is that it cannot produce detent torque. Detent torque is the amount of torque the motor produces when the rotor coils are not energised. You can feel the 'cogging' from the detent torque on the permanent and hybrid motors when turning the shaft by hand with no power applied. This type of stepper motor is not commonly used and has limited availability. They tend to be used in very specific markets – you may well find one driving your new washing machine drum. They will not be described further in this series.

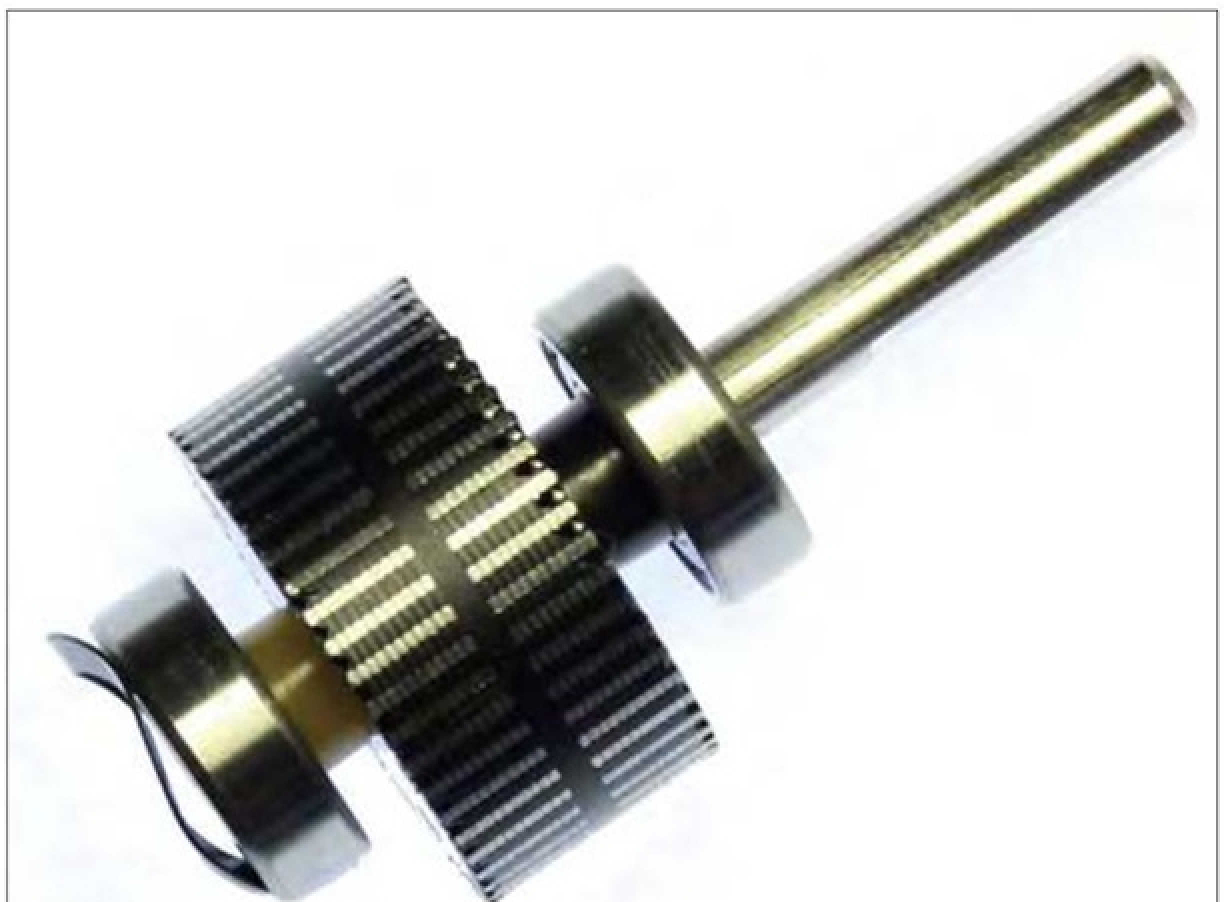


Fig.3. A NEMA 17 hybrid stepper motor rotor from a 200-step-per-revolution motor with its two rotor sections, each with 50 teeth. Look closely and you should see the two rotor sections' teeth are offset from each other by half a pitch (3.6°) effectively making a 100-tooth rotor.

Hybrid stepper motors

Last, we have hybrid motors, which use elements of both permanent magnet and variable-reluctance stepper. Hybrid stepper motors consist of a permanent magnet rotor with radial soft iron teeth (see Fig.3), and a multi-pole wound stator with corresponding teeth (see Fig.4). They are perhaps the most common type of stepper motor as they offer better step resolution, holding torque and speed, but can be more expensive than the other types. Perhaps the most common hybrid stepper motors have 200 steps per revolution which is 1.8° (360°/200). 400-step motors are available which give a step size of 0.9° (360/400). Fig.5 and Fig.6 show a NEMA 17 400-step rotor and stator respectively. The drive electronics tends to be similar for

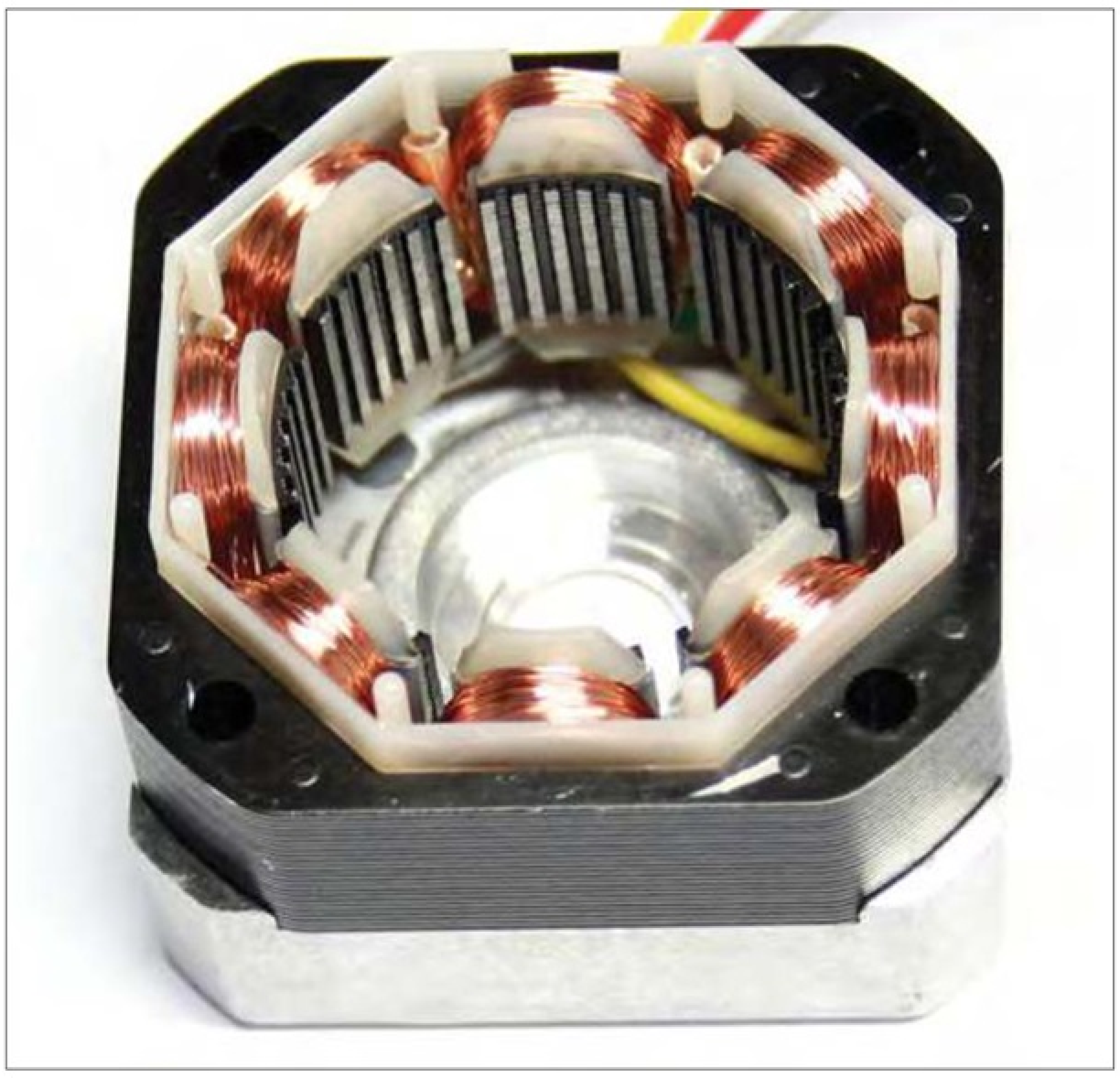


Fig.4. A NEMA 17 2-phase hybrid bipolar stepper motor stator with the rotor (see Fig.3) removed. Each of the 8-poles has a winding, the windings on poles 1, 3, 5 and 7 are wired in series to form phase 'A', similarly 2, 4, 6 and 8 are series wired for phase B. The controller sequentially energises the phases which attract the teeth on the rotor until some of the rotor teeth are aligned with some of the 48 stator teeth (two fewer than the rotor). The other teeth are out of alignment by ¾, ½ or ¼ of a tooth pitch, so with the next step, the rotor will move ¼ of a 7.5° step or 1.8°, which equals 200 steps per revolution.

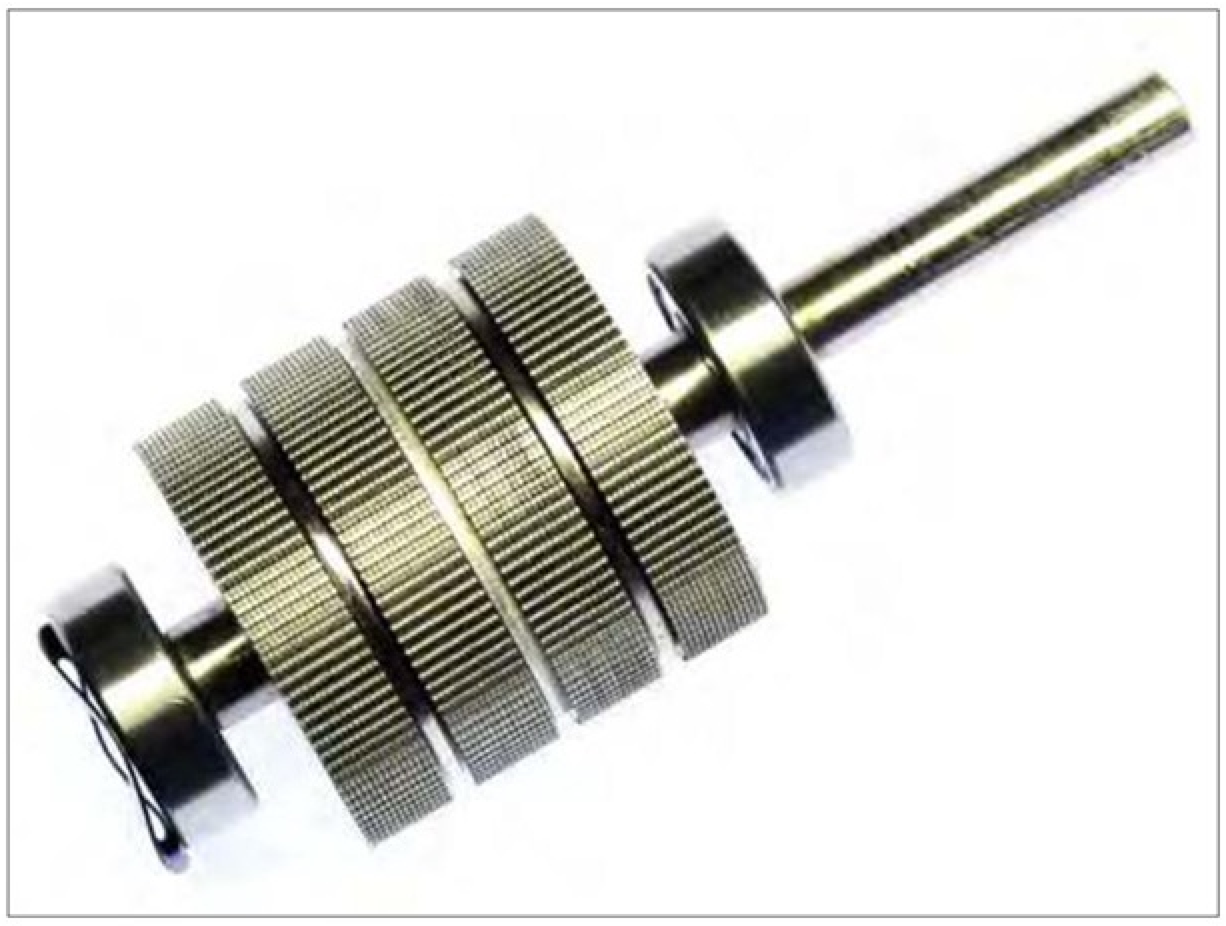


Fig.5. NEMA 17 400-step bipolar motor rotor, note the increased number of teeth compared to the 200-step stator in Fig.3.

both permanent and hybrid stepper motors, and can effectively increase the number of steps per revolution many fold with techniques that will be discussed in a later article.

How a hybrid stepper motor works

Most tutorials on the Internet use a very simplified view of a stepper motor that does not really help in understanding how these 200-step motors actually function. Describing this in the necessary detail in these articles would be less effective than the animated graphics on the excellent explanation in the Nanotec video, which can be viewed at: **https://youtu.be/Ew6eVGnj7r0** (recommended viewing for those who would like a better understanding of what happens in a hybrid stepper motor).

Unipolar and bipolar windings

When looking at permanent or hybrid stepper motor, you may well notice in the description they are referred to as 'unipolar' or 'bipolar'. This describes the motor wiring winding configuration and are classified as either unipolar or bipolar, see Fig.7.

4-phase unipolar motors consist of two windings each with a centre tap. How the centre taps are brought out can be individually (6-lead, see Fig.7c) or as a common (5-lead Fig.7b). 2-phase bipolar motors do not have a centre tap, just the ends of the windings are brought out (4-lead Fig.7a). 6-lead unipolar motors can also be used as a bipolar motor by not using the two centre tap leads. There is a third arrangement called 'bifilar', where instead of winding each coil with a single wire, two wires are wound in parallel with each end brought out (8-lead Fig.7d). This allows the motor to be driven in either unipolar or bipolar by connecting the windings in series or parallel.

Unipolar motors (Fig.8) are easier to drive in that the driver circuit does need to reverse the current through the winding making the circuitry much simpler to implement. The bipolar motor requires a full 'H' bridge driver circuit to allow the current to be reversed and thus reverse the magnetic field of the windings. We will discuss driver circuits in a later issue. You may be wondering why not always use the simpler approach to drive unipolar motors. It's simply that unipolar motors suffer from reduced torque as only half of each winding is energised at a time, unlike a bipolar where the whole winding is energised. Unless you are building your stepper driver from discrete components, bipolar is the way to go as many semiconductor manufacturers now offer dedicated H-bridge driver chips.

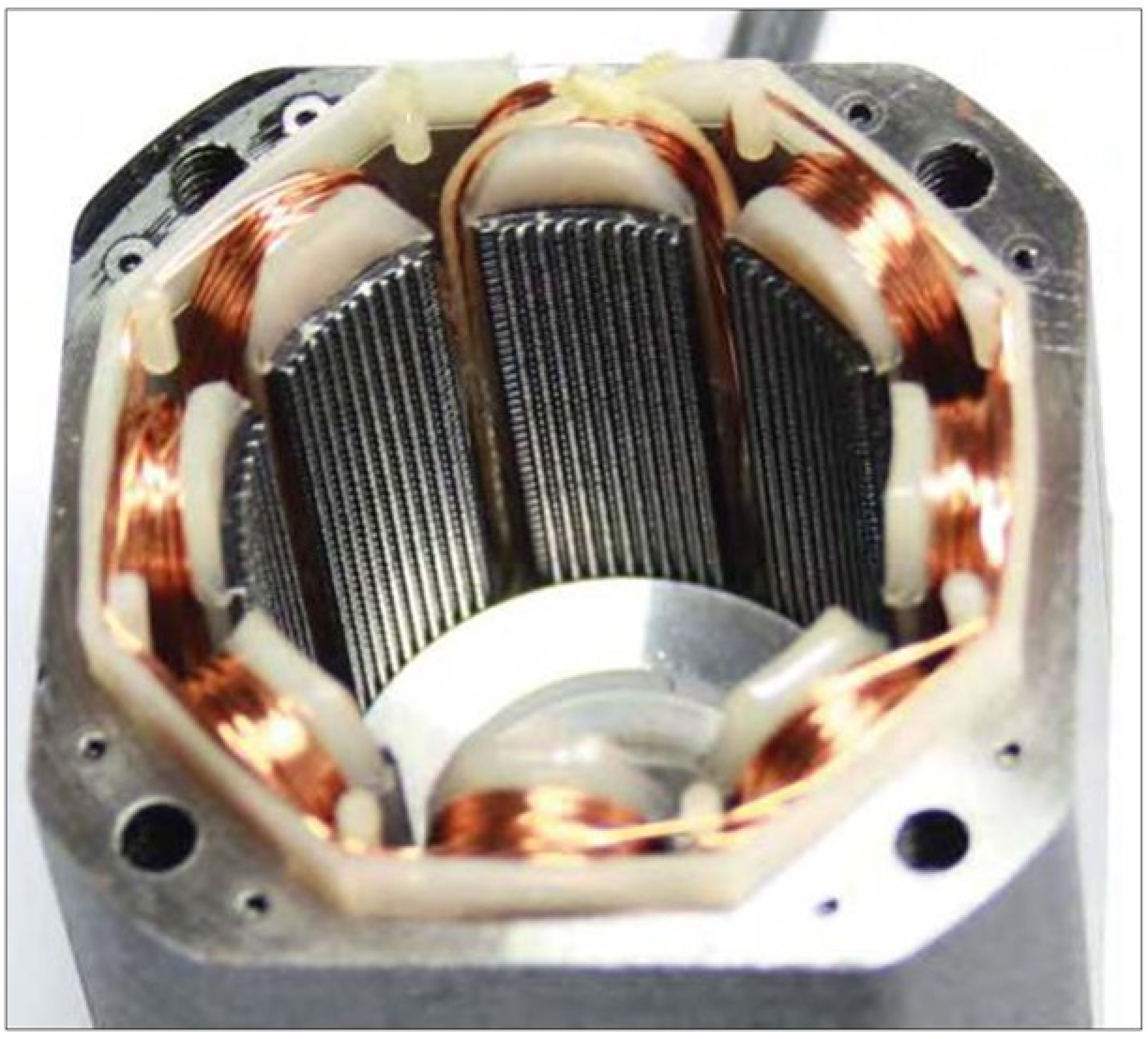


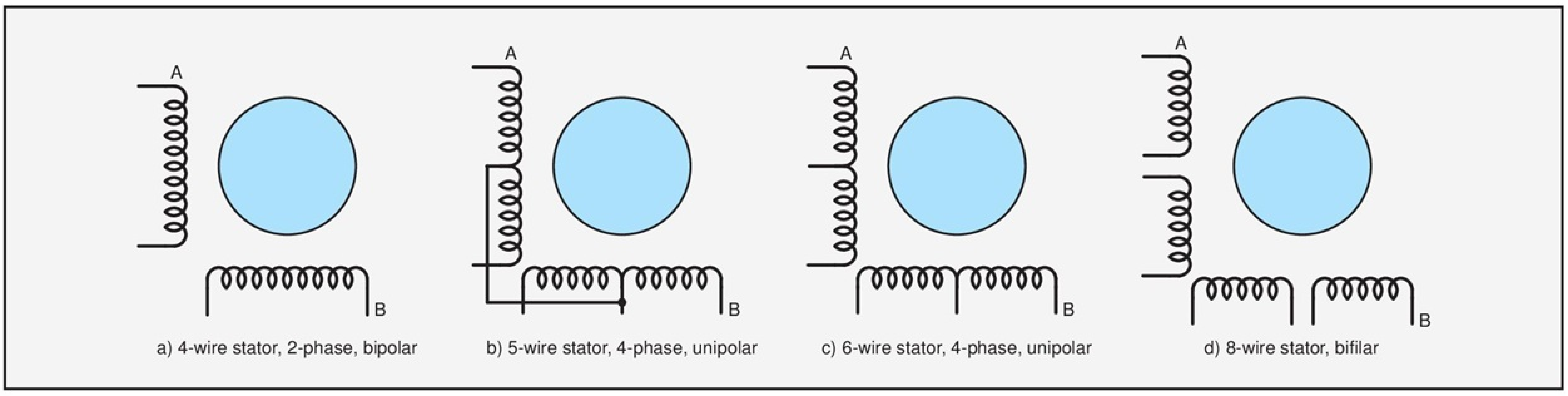
Fig.6. NEMA 17 400-step bipolar motor, like its stator, it also has an increased number of teeth compared to the 200-step stator in Fig.4.

Hybrid motor sizes

Of the three stepper motor types, we now know that the hybrid variety is the most commonly available and bipolar is the most common drive method. Hybrid stepper motors are available in various physical frame sizes and are identified by a NEMA number. 'NEMA' is an acronym for the National Electrical Manufacturers Association, a US-based body representing over 300 electrical equipment manufacturers from various market sectors. The NEMA standard is based on the inch, and despite this it is the most common frame size reference. There is also an IEC metric frame size standard, but for the hobby market most retailers use the NEMA classification.

The NEMA number is related (approximately) to the width of the square face of the stepper motor (Table 1). There is also a NEMA 24 size, which while not a formal standard, mounts like a NEMA 23 but is slightly wider, which allows for a larger rotor and stator, and thus more torque.

The NEMA 17 is widely used by hobbyists, especially those building 3D printers. For a given NEMA motor size, the body length can also vary, the greater the body length, the greater the



a) 4-wire stator, 2-phase, bipolar
b) 5-wire stator, 4-phase, unipolar
c) 6-wire stator, 4-phase, unipolar
d) 8-wire stator, bifilar

Fig.7. Stepper motor wiring configurations; the 4-wire type is most commonly used.

FD819MX77

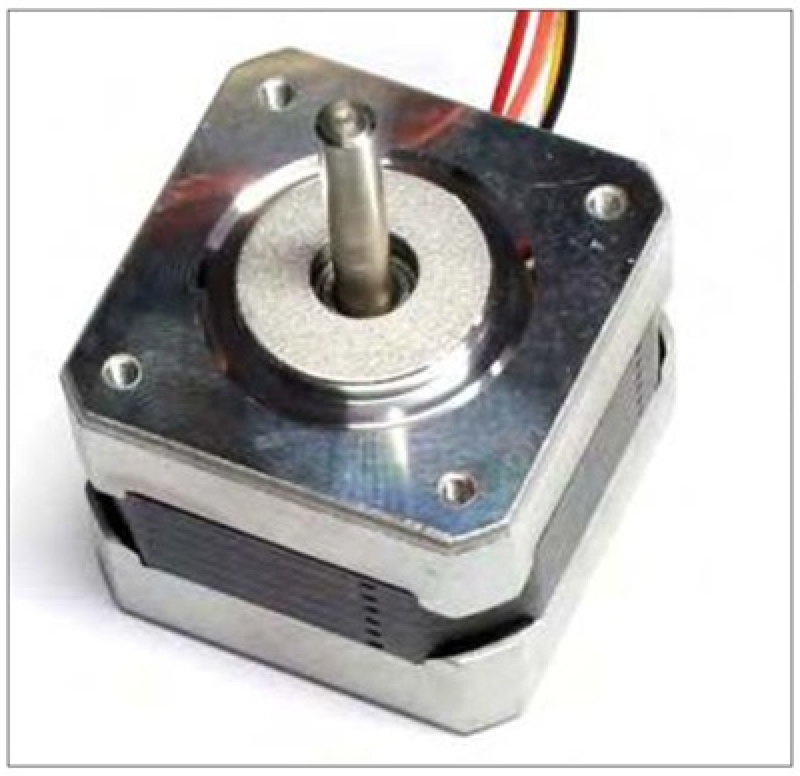


Fig.8. Example of a NEMA 17 unipolar stepper motor.

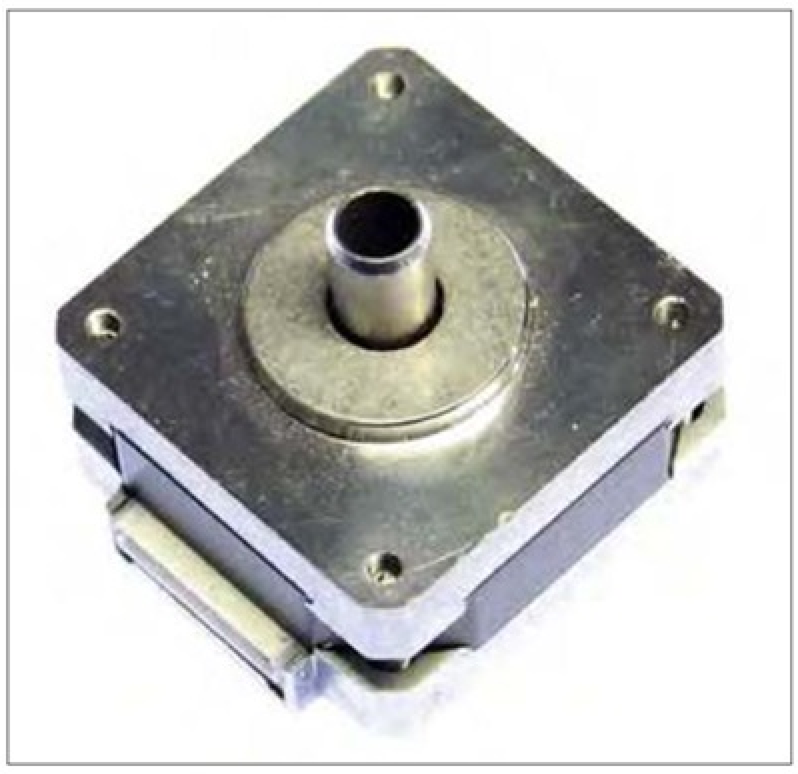


Fig.9. NEMA 17 motor with a hollow shaft – Technobots used these for a client who needed to continuously rotate in one direction a very bright LED light panel. The motor shaft drove a slip ring, which in turn drove the LED panel around. The slip ring wires were passed through the hollow shaft to the LED driver such that the wires never twisted.

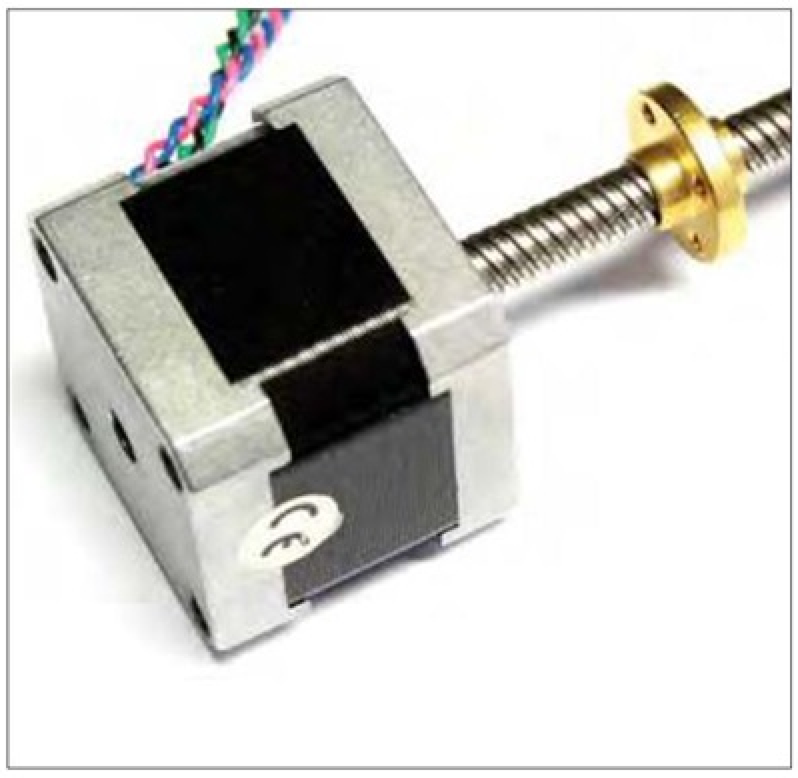


Fig.10. NEMA 17 bipolar stepper motor with integral T4 lead screw driving a brass 'nut'.

motor torque. Other sizes of stepper motor are used in commercial applications, both smaller and larger than the NEMA range below. While the NEMA number relates to the physical size, other mechanical and electrical specifications can and do differ. Two NEMA-sized motors may look identical, but their electrical performance can be very different and they are not necessarily interchangeable.

Various motor shafts are also available for stepper motors, round shafts, 'D' shafts, hollow shafts (see Fig.9), even lead screw shafts where rotational to linear movement is required (Fig.10).

Next month

In November's article, we will look at choosing a stepper motor and how to identify the leads on that surplus stepper motor you extracted from a defunct piece of equipment.

Table 1: Relationship between NEMA frame size number and the face width.

	NEMA 8	NEMA 11	NEMA 14	NEMA 17	NEMA 23	NEMA 34	NEMA 42
Motor face size (inches)	~0.8	~1.1	~1.4	~1.7	~2.3	~3.4	~4.2

Technobots
Supporting your projects since 2001

Mechanical
Bearings, Gears, Sprockets, Pulleys, Shaft, T-slot extrusion

Motors
Stepper motor drivers, DC controllers, Miniature and Geared Motors

Electronics
Passives, semiconductors, breakout boards, sensors, connectors and leads

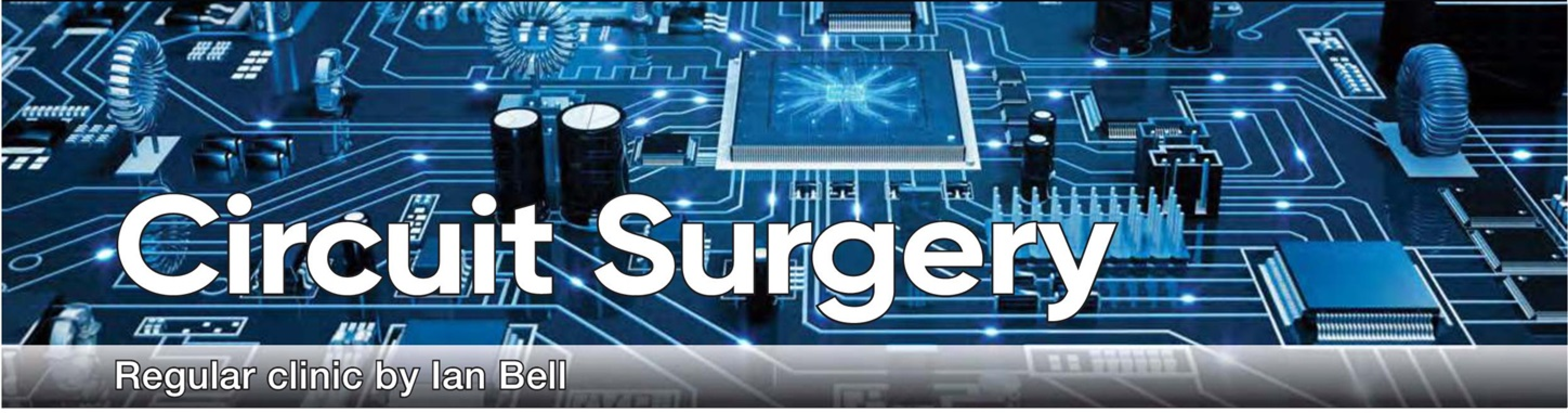
MakerBeam
Pololu Robotics & Electronics
FingerTech Robotics
MFA/Como Drills
MaxBotix

Order online at
www.technobotsonline.com

Follow us at
facebook.com/TechnobotsUK

Unit 29, Webb Ellis Business Park
Rugby, Warwickshire
CV21 2NP

Tel: 01788 877 400



Circuit Surgery

Regular clinic by Ian Bell

Understanding bipolar junction transistors – Part 4

This month, we conclude our look at the different bipolar transistor circuit configurations – common-emitter, common-base and common-collector – inspired by questions on this topic on the EEWeb forum (**http://bit.ly/eeweb-pe**). Previously, we've looked at the common-emitter and common-base configurations – specifically the basic single transistor amplifier circuits that use these configurations. This month, we'll focus on the common-collector amplifier, also known as the 'emitter follower'. To support the discussion on specific circuit designs we have also covered some general topics, including biasing, small-signal modelling and high-frequency behaviour of the transistor. Last month, we also looked at input impedance and here we return to this, along with output impedance, in the context of the common-collector amplifier.

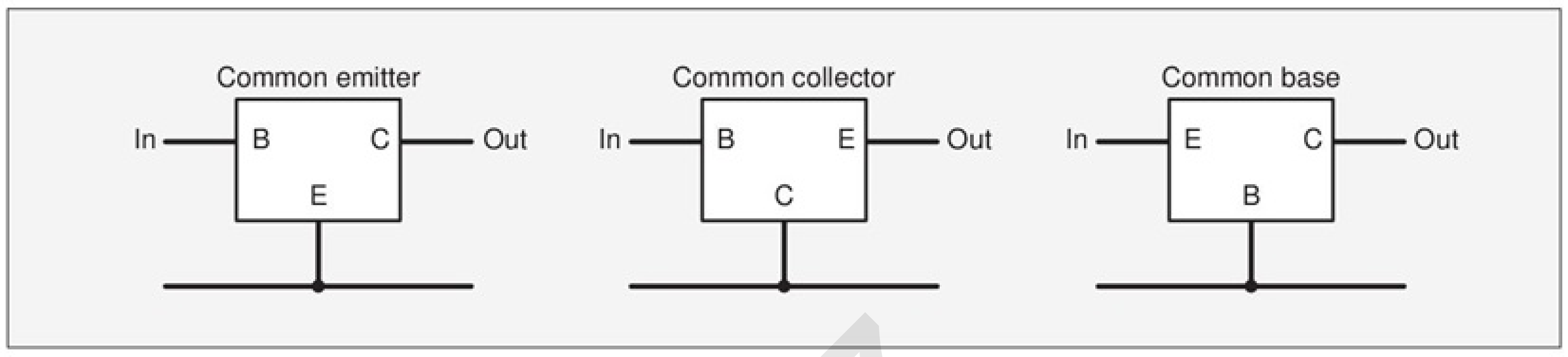


Fig.1. Concept of transistor configurations. These apply to both NPN and PNP. Actual circuits require additional components.

Common collector

Looking at Fig.1, we see that unlike the common-emitter and common-base circuits discussed previously, the output from the common-collector configuration is from the emitter, not the collector. Fig.2 shows the basic common-emitter amplifier without bias circuitry. As with the other circuits, we connect a resistor from the configuration's output terminal to the supply, but due to the polarity of the transistor we need to connect the output resistor between the emitter and ground. It serves the same purpose as the collector resistor in the other configurations – connecting power to the transistor and converting the transistor's output current to an output voltage. The common connection – the collector – is connected directly to the power supply. This might seem odd as we usually associate the common connection for a signal with ground, but recall from our previous discussion on small-signal models that as far as the signal is concerned the power supply is like a short circuit. So from a signal perspective, the collector is directly connected to ground in the circuit in Fig.2.

Simulation files

Most, but not every month, LTSpice is used to support descriptions and analysis in *Circuit Surgery*.
The examples and files are available for download from the *PE* website.

Like the other configurations, we need to supply bias for the circuit to work. Again, we can use a potential divider at the base (see Fig.3) and follow similar arguments to those used with the other circuits to arrive at suitable component values. We chose a bias current for the emitter (I_E) as the operating current, taking factors such as optimal transistor performance and power consumption into consideration. We choose a no-signal output (emitter) voltage (V_E), and half the supply gives the largest potential output amplitude, but other values can be used if desired. The chosen V_E and I_E set the emitter resistor value ($R_1 = V_E/I_E$). The transistor's base-emitter voltage (V_{BE}) to emitter current relationship – the Ebers-Moll diode equation that we discussed in *Part 2* – tells us the required V_{BE} based on our chosen operating I_E.

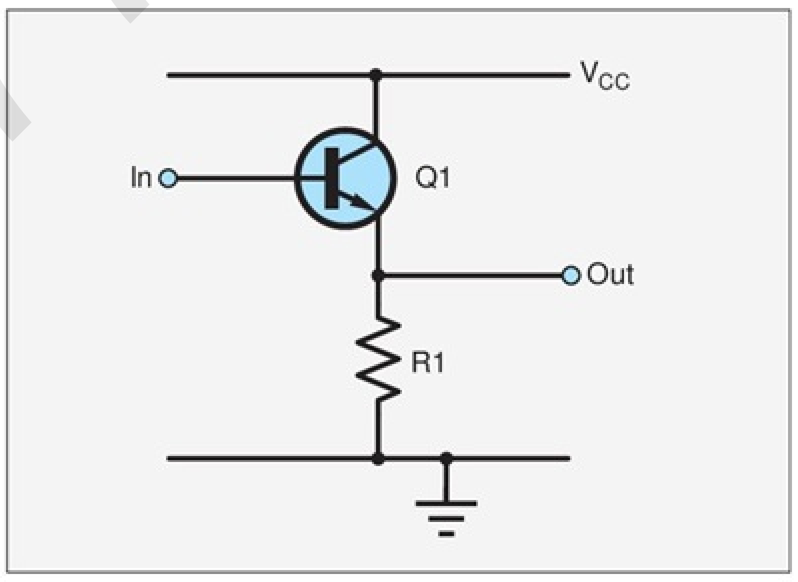


Fig.2. Basic common-collector amplifier without bias circuitry.

Bias calculations

If we use similar values to the common-emitter circuit discussed in *Part 2*, we choose I_E = 1mA and with V_{CC} = 12V, and a no-signal emitter voltage of $V_{CC}/2$ = 6V, then we get R_1 = 6kΩ. In *Part 2* we discussed the non-linear relationship between V_{BE} and I_E in detail, and for the example transistor used (BC547B), found that a V_{BE} of 0.635V was required for an emitter current of 1mA. This is based on the model provided with LTspice and of course, as we have discussed, will vary with individual transistors.

Having found our required V_{BE} we can select suitable resistors for the bias potential divider. With the emitter at 6V we need 6.635V at the base for V_{BE} = 0.635V.

Many resistor combinations are suitable, but we should not use values that are too small because of power consumption and the effect on input impedance. The values should also not be too large – we should have a current in the bias resistors more than ten times the bias base current. Again, this was discussed previously. We can choose a reasonable total resistance (R_T), say 30kΩ, and use the required ratio of supply-to-base voltage to find the grounded resistor (R_3 in Fig.3), that is $R_3 = (V_B/V_{CC})R_T$ = (12/6.635) × 30kΩ = 16.6kΩ. The other resistance is simply this value subtracted from the chosen total (here, 13.4kΩ).

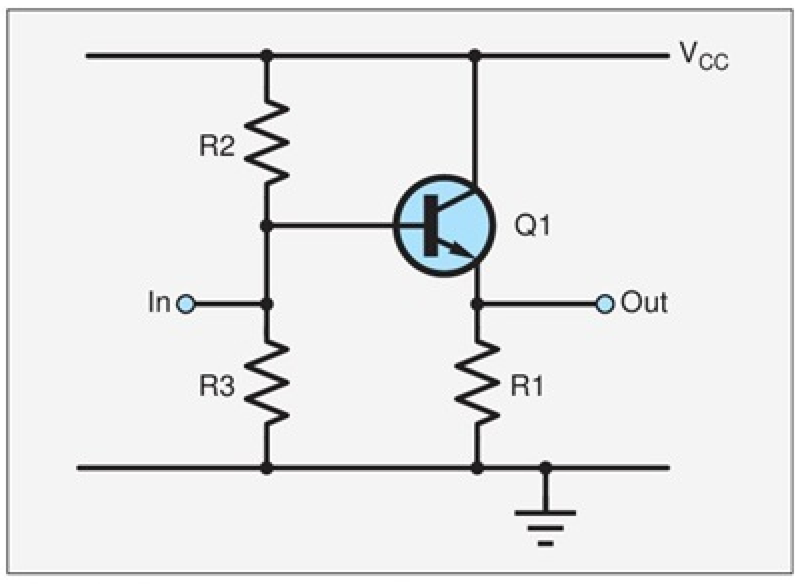


Fig.3. Common-collector amplifier DC bias circuitry.

FD819OMX77

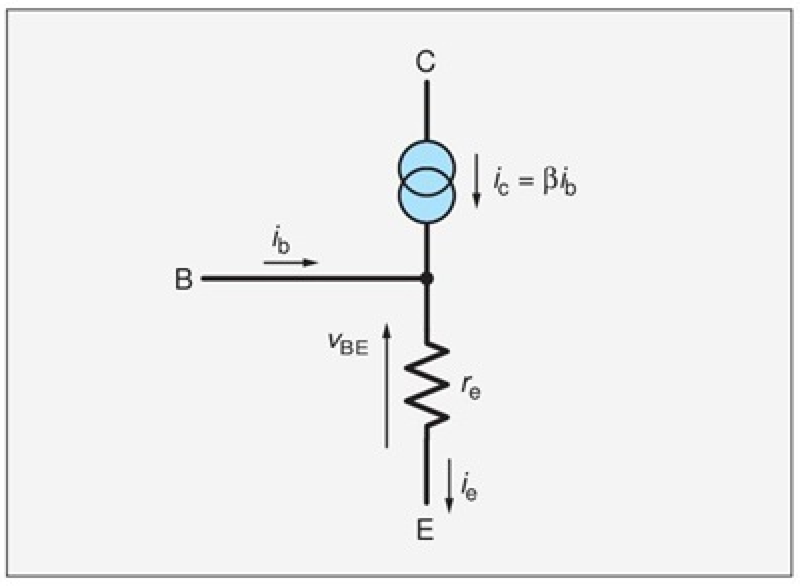


Fig.4. Transistor small-signal model with r_e.

Variations in individual transistors and changes in circuit conditions, such as temperature mean that a carefully calculated bias set up may differ significantly for individual transistors, or vary during operation. To overcome this we can apply feedback to stabilise the bias; again, we introduced this idea for the common-emitter circuit, where we added a resistor in the emitter circuit. The common-collector circuit already has an emitter resistor, so the circuit in Fig.3 has stabilised bias without needing extra components. Another issue with the bias components is their potential impact on input impedance – we will return to discuss this later.

Analysis

We can analyse the circuit using a small-signal model, as we have done with other circuits we have looked at. In this case, we will use the model with r_e rather than r_{be} (see Fig.4) – as we did for the common-base circuit last month. To keep things as simple as possible we will ignore the bias components and substitute the model for the transistor in Fig.2, adding a signal source. We also use an ideal input source with zero source resistance. This gives the small-signal model shown in Fig.5 – note the power supply and ground connected together, as discussed earlier. We can redraw this schematic with an alternative layout, as shown in Fig.6.

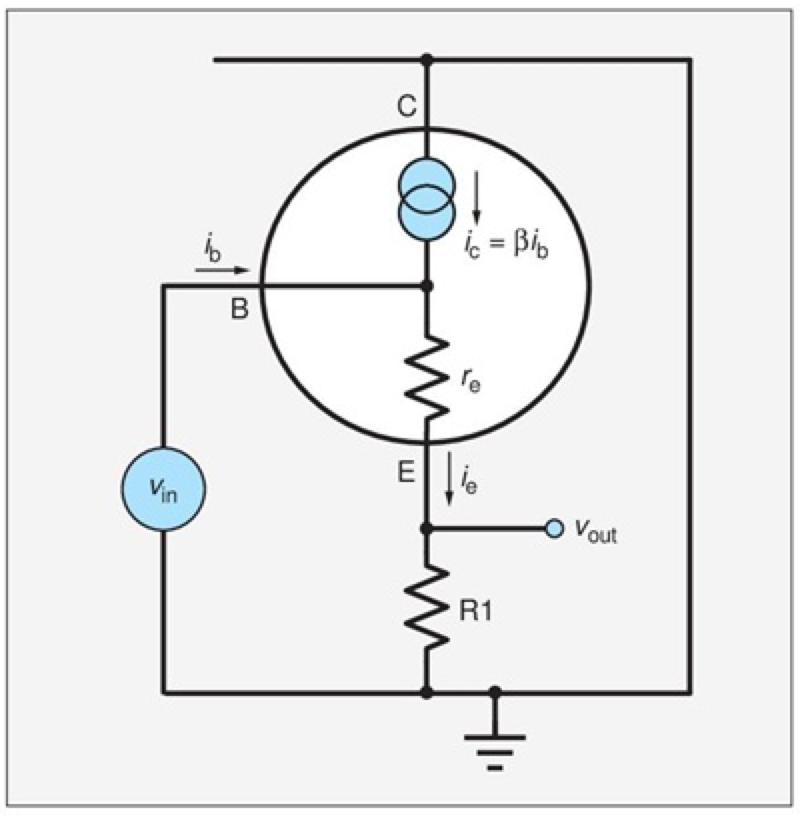


Fig.5. Small-signal model for common-collector amplifier ignoring bias components.

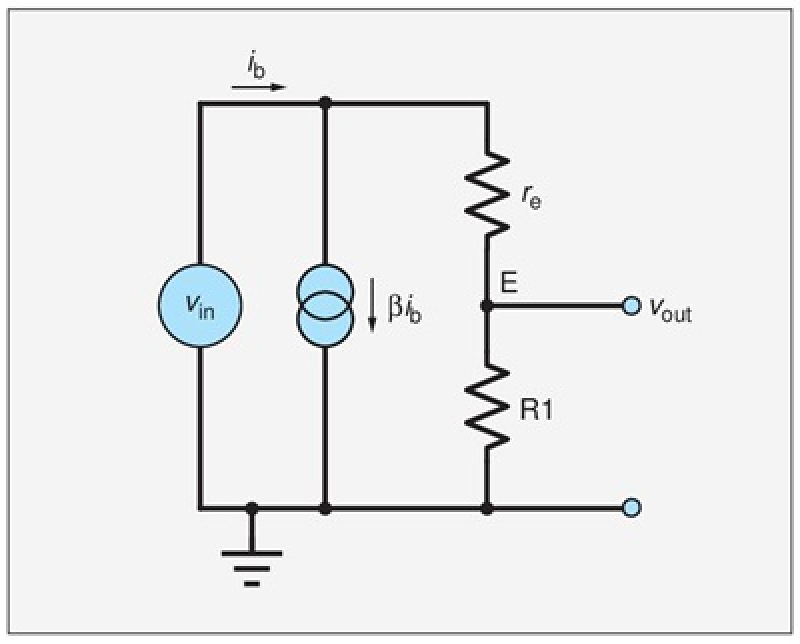


Fig.6. Fig.5 redrawn with alternative layout.

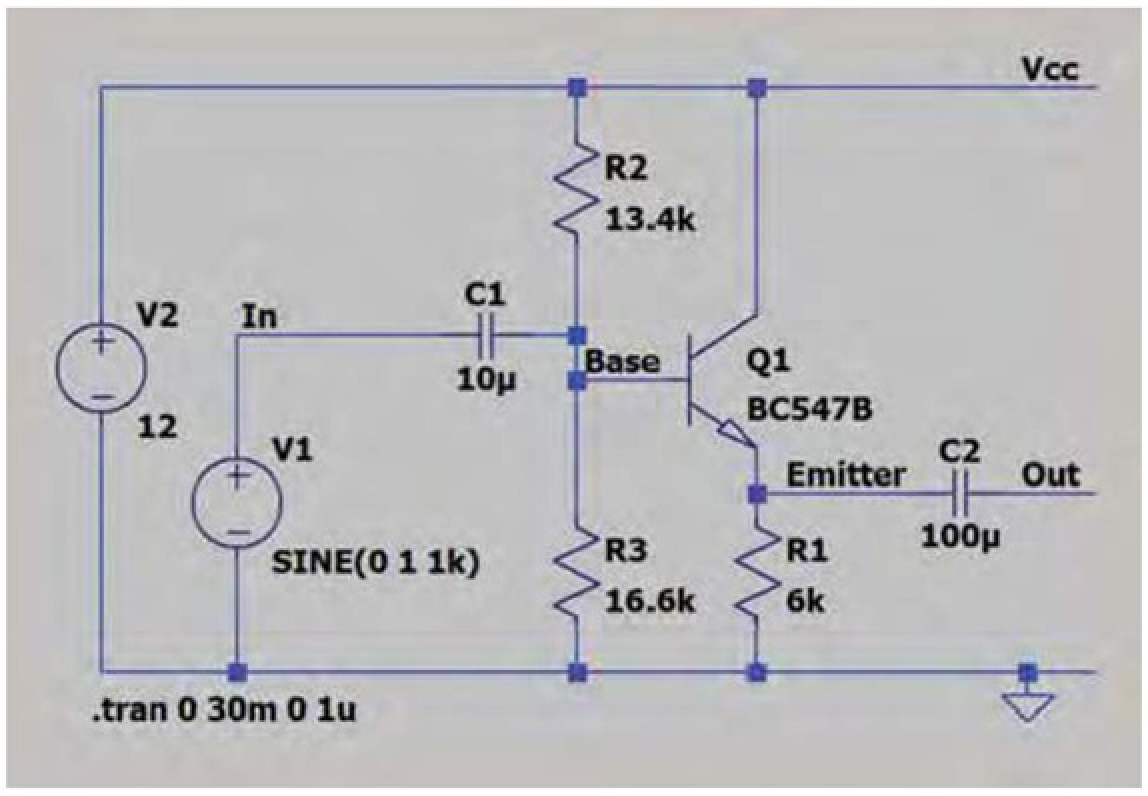


Fig.7. LTspice schematic for a basic common-collector amplifier.

Referring to Fig.6, it is reasonably straightforward to see that r_e and R_1 form a potential divider, so we have:

$$V_{out} = \frac{R_1}{R_1 + r_e} v_{in}$$

So the gain is:

$$\frac{R_1}{R_1 + r_e} \approx 1$$

This value is usually typical close to (and always slightly less than) unity because in a typical circuit r_e will be significantly smaller than R_1. The value or r_e is approximately $25/I_E$ at room temperature, so with an operating current of 1mA r_e is around 25Ω. This indicates a gain of 6000/(6000+25) = 0.996 with R_1 equal to 6kΩ as selected above – close to unity, as expected.

The model circuit used is simplified – we have ignored the input (v_{in}) source resistance, the bias components and the transistor's output resistance. Including these components results in more complex algebra for the circuit analysis, but produces fundamentally the same result: a gain close to unity – at least under the assumption that the transistor gain is typically 'large' (eg, 100 or more) and that the resistance of the bias components is significantly larger than the source resistance.

Simulation

Fig.7 shows an LTspice schematic for a basic common-collector amplifier, with input and output decoupling capacitors, and component values as discussed for Fig.3 above. The results of simulating the circuit are shown in Fig.8 – here we see that the output voltage amplitude is very close to the input voltage and is in phase with the input. The common-collector amplifier is non-inverting, unlike the common-emitter layout. The fact that the output voltage signal is very similar to the input leads to the name 'voltage follower' commonly being used for this circuit.

If we zoom in very close to a peak in waveforms, as in Fig.9, we can see that (as expected) the amplitude of the output is slightly less than the input. Measuring this simulation with the waveform cursors indicates a gain of 0.994, which given the amount of simplification used is acceptably close to the value calculated earlier. Fig.10 shows the voltages directly at the base and collector. These are the same signal waveforms as in Fig.8, but include the DC bias voltages around the

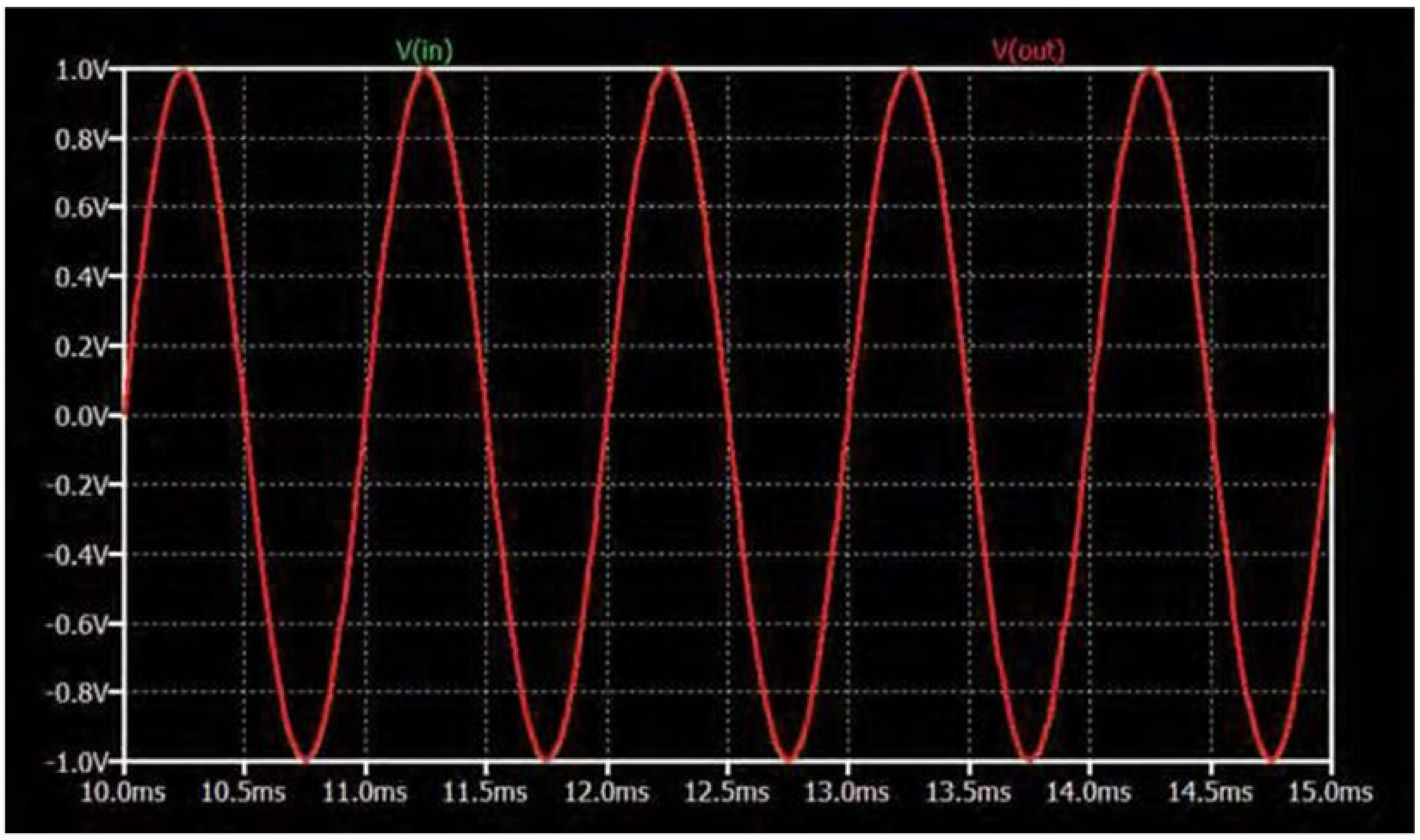


Fig.8. Simulated input and output waveforms for the circuit in Fig.9. These voltages are almost equal.

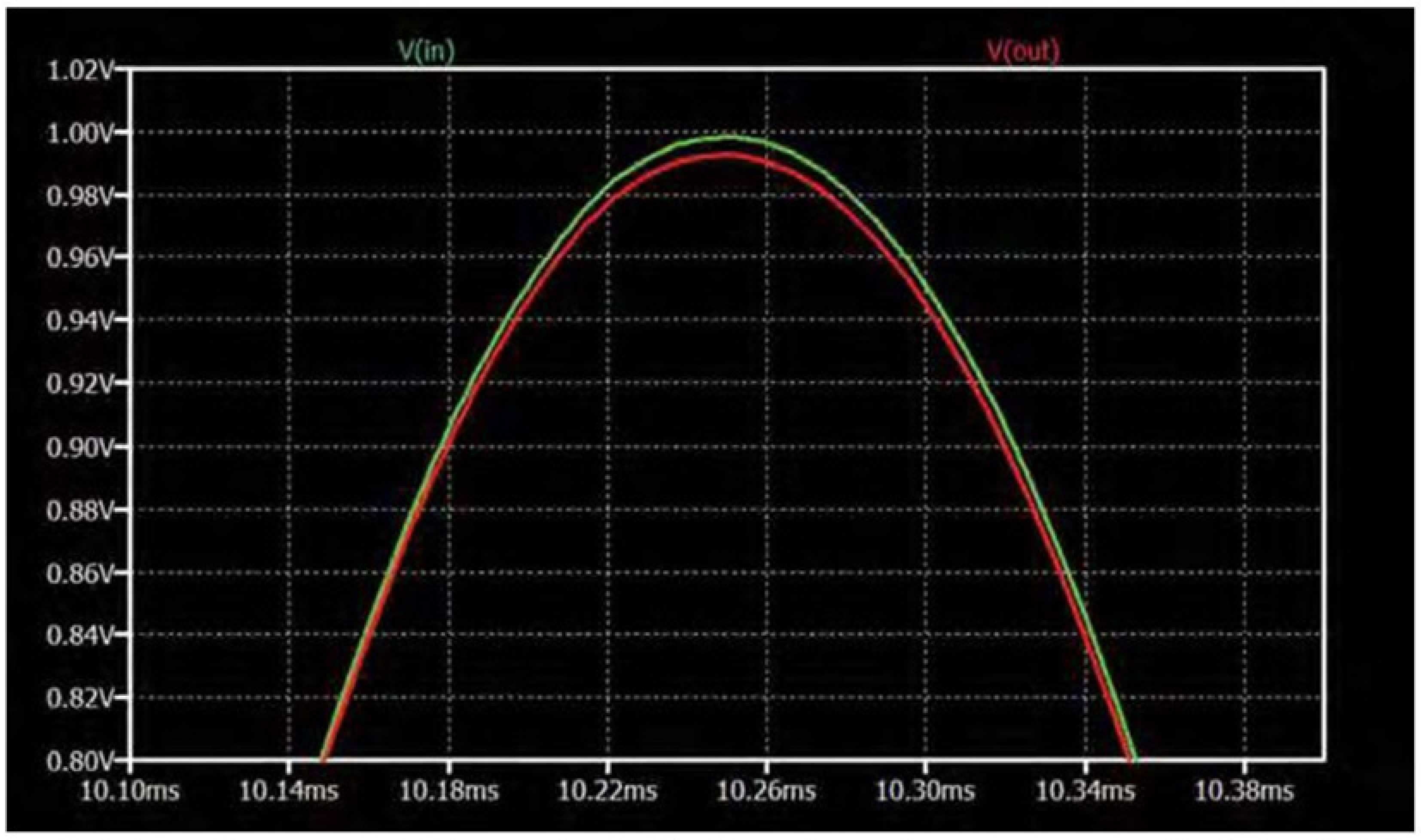


Fig.9. Zoom in on one of the peaks in Fig.9 to show that output amplitude is slightly smaller than the input.

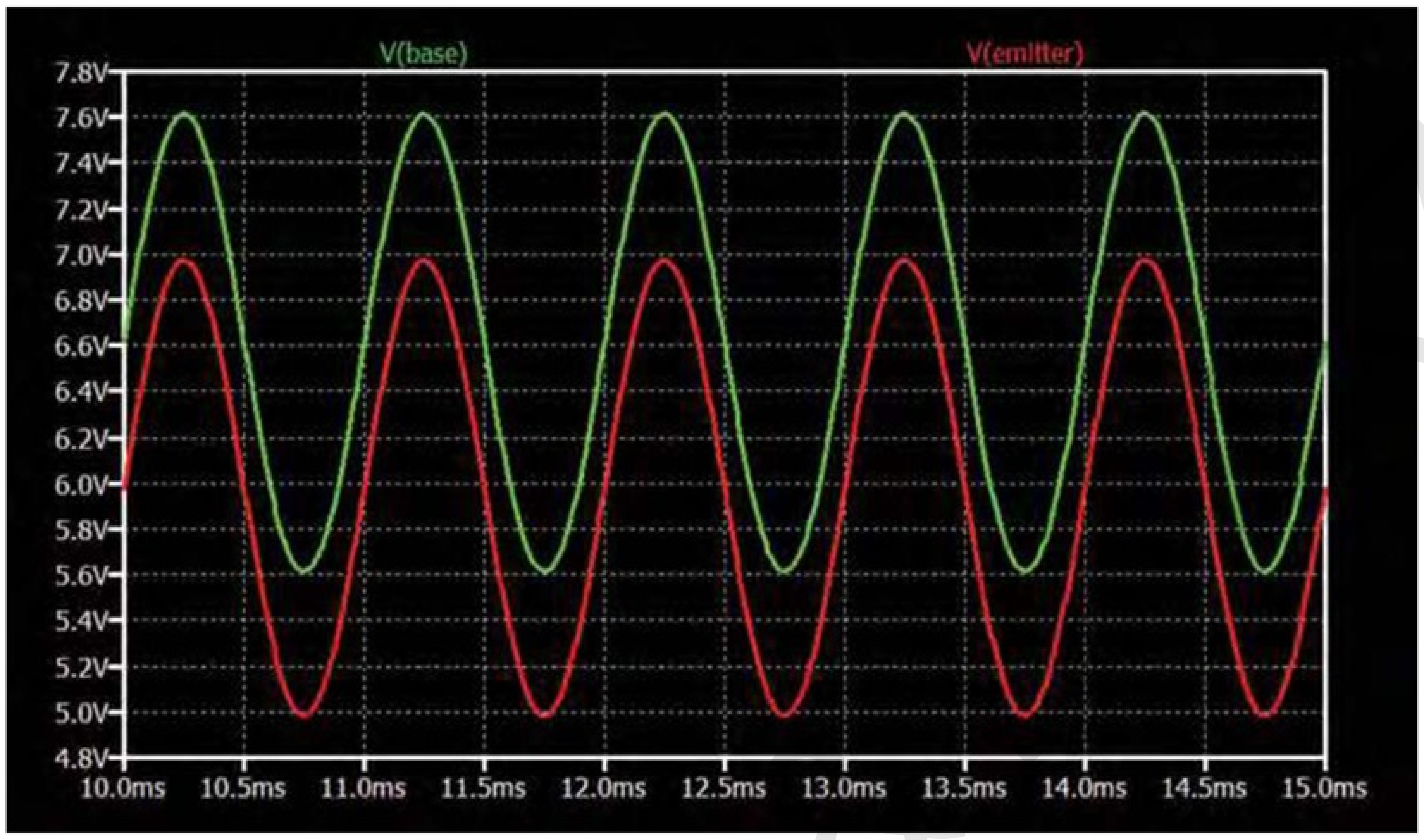


Fig.10. Simulated waveforms from Fig.7 directly at the base of an emitter. This shows the signals together with the bias voltages around the transistor.

transistor, which are blocked by the coupling capacitors. We see that the DC levels of the signals at the emitter and base have a constant 0.66V DC difference between them – the V_{BE} voltage set by the biasing.

Input and output impedance

On first thoughts, an amplifier with unity voltage gain may not seem very useful; however, this is not the whole story. The common-collector amplifier has a current gain much larger than unity – it can deliver the output signal at a much higher current level than it takes from its input. This leads us to a discussion of the input and output impedance of amplifiers. Last month, we looked at amplifier input impedance in the context of how it might vary with frequency, potentially reducing signal amplitude at high frequencies. We used a generic circuit like the left-hand side Fig.11 to represent the connection between the signal source and amplifier input. The same form of circuit model can be used for the output of the amplifier connected to a load – as seen on the right-hand side of Fig.11. Overall, Fig.11 is a model of the source, amplifier and load together.

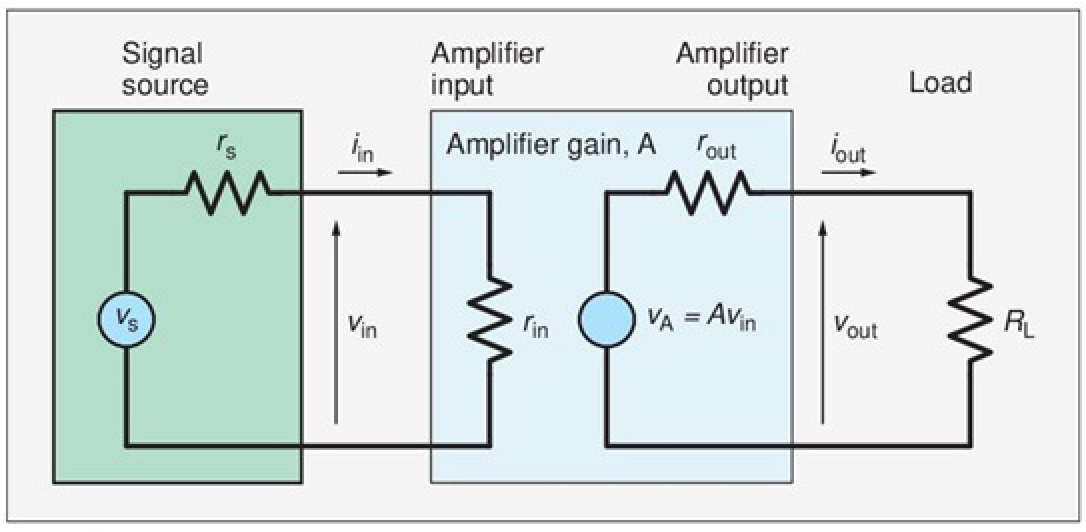


Fig.11. Generic circuit for a signal source connected to an amplifier driving a load.

The idea of representing the output of a complex circuit, such as an amplifier or filter, by a single voltage source and resistor (or impedance) is based on Thévenin's theorem, which is named after Léon Charles Thévenin (1857–1926), who was a French telegraph engineer. Thévenin's theorem states that a linear electronic circuit that comprises any combination of voltage sources, current sources and resistors, with two output terminals, is electrically equivalent to a single voltage source and a single series resistor – for example, the amplifier output voltage (v_A) and output resistance in Fig.11. The theorem also applies to circuits in which the sources are AC (sinewave), all at the same frequency, and the other components are impedances (resistance, capacitance, inductance).

Finding Thévenin

To obtain the Thévenin equivalent for any suitable circuit we find the open-circuit output voltage and then the short-circuit output current. The open-circuit voltage is used as the equivalent source voltage and the open-circuit voltage divided by the short-circuit current gives the source resistance.

In Fig.11 the source is also modelled in the same way – it could be the Thévenin equivalent of a previous amplifier stage, or it could be some physical signal source, which can also be modelled as a voltage source and an associated source impedance. Use of pure resistance for input and output impedance may be a simplification in both cases (see last month's discussion), but it is sufficient for our purposes here.

If the amplifier in Fig.11 has a much higher input resistance (r_{in}) than the source's output resistance (r_s) then the potential divider formed by r_s and r_{in} will produce a voltage very close to the applied voltage; that is, v_{in} will be very close to v_s and we can assume $v_{in} = v_s$. If the amplifier has unity gain ($A = 1$) then we have $v_A = Av_{in} = Av_s = v_s$.

We can find the Thévenin equivalent of the combination of the source and amplifier. The open-circuit output voltage (ie, with R_L disconnected) is v_A (so equal to v_s). The short-circuit output current (with $R_L = 0$) is simply v_{out}/r_{out}, so we just get r_{out} as the Thévenin source resistance. Therefore, the Thévenin equivalent of the circuit in Fig.11 is as shown in Fig.12. The key thing to note here is that we have the same source voltage, but with an output resistance of r_{out} instead of the original

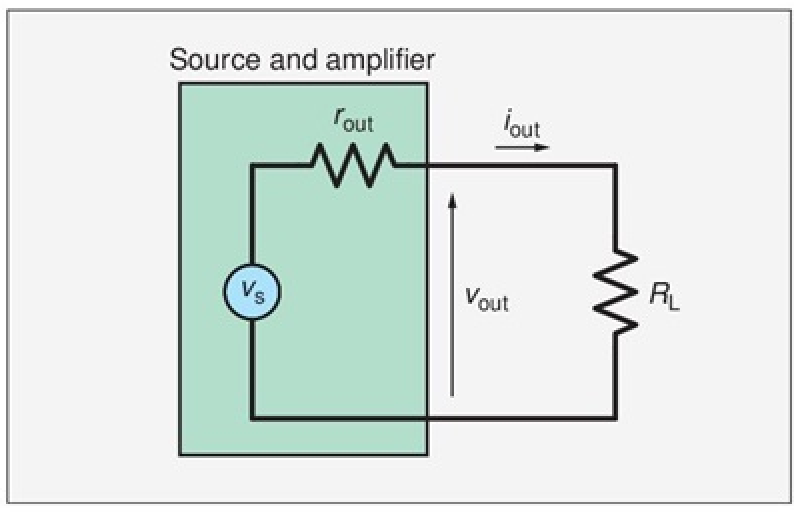


Fig.12. Thévenin equivalent of the circuit in Fig.11 for amplifier with unity gain and very high input resistance.

r_s. On the assumption that we have an amplifier with low r_{out} (as we do with a common-collector circuit), another way to describe this to say that that a buffer amplifier effectively lowers the output impedance of the signal source applied to it.

A common use of emitter-follower circuits is the output stage of multiple-stage (multi-transistor) amplifiers. Typically, the first stages provide high voltage gain and the common-collector circuit facilitates delivering the amplified voltage at relatively high current to the load. Power amplifiers – for example, those used to drive loudspeakers – can be based on the emitter follower. However, in these circuits two transistors are employed to cover the negative and positive half cycles of the waveform separately, which leads to additional complexity in the biasing, which we will not cover here.

Input resistance

Using our small-signal model circuit (Fig.5) we can get some idea of the actual input and output resistance of a common-collector circuit. Starting with input resistance, this is by definition:

$$r_{in} = \frac{v_{in}}{i_{in}} = \frac{v_{in}}{i_b}$$

For the transistor, the emitter current is the sum of the base current (i_b) and collector current (equal to βi_b), that is:

$$i_e = i_b + \beta i_b = (1 + \beta) i_b \approx \beta i_b$$

In which we can approximate $(1 + \beta) = \beta$ on the assumption that β is much larger than 1, from which we get:

$$i_b = i_e / \beta$$

Using this in the r_{in} equation we can write:

$$r_{in} = \frac{v_{in}}{i_e} \beta$$

But we have a unity-gain amplifier ($v_{out} = v_{in}$), so:

$$r_{in} = \frac{v_{out}}{i_e} \beta$$

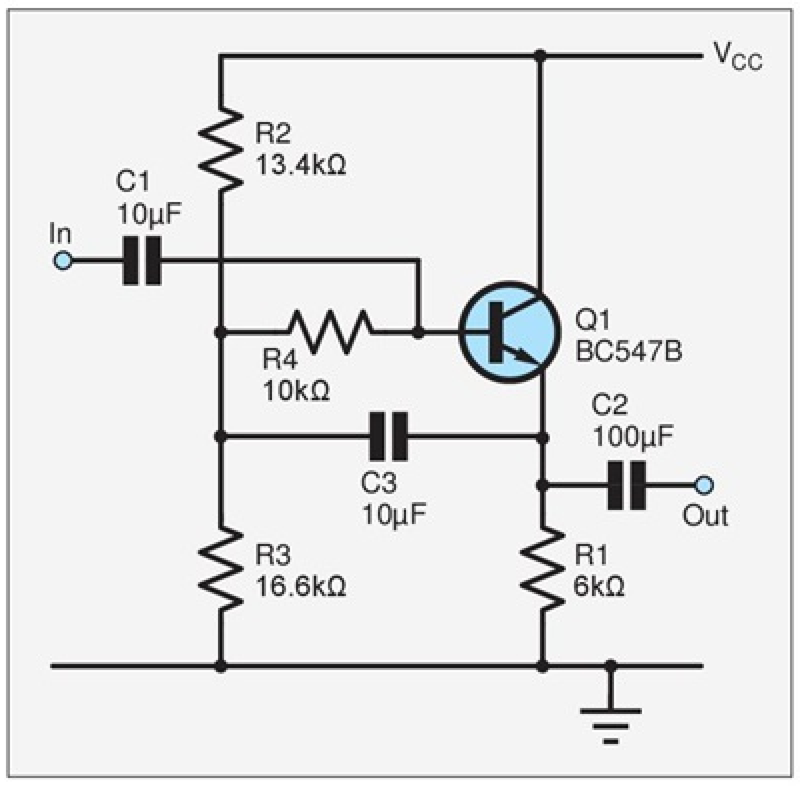


Fig.13. Common-collector amplifier with bootstrap bias.

But, looking at Fig.6 we see that (using Ohm's law) $R_1 = v_{out}/i_e$, so:

$$r_{in} = R_1 \beta$$

The input resistance of the circuit is the emitter (or load) resistance multiplied by the transistor gain. As before, our analysis has been simplified by ignoring the bias components. In the small-signal model both bias resistors are connected between the base and ground, so they are effectively in parallel with the input resistance we have just calculated. This has a significant effect. For example, for the circuit in Fig.8, with a β of 300, we have a potential r_{in} of 6kΩ × 300 = 1.8MΩ, but this will be greatly reduced by the less than 10kΩ parallel combination of the bias resistors.

We can use different bias arrangements to overcome this problem. A simple approach is to use a single resistor between V_{CC} and the base (R_2 in Fig.9, with no R_3 present). The value is calculated using the required base current. For example, for an operating current of I_E = 1mA and a β of 300, we require a base current of 1mA/300 = 3.33µA. With a required base voltage of 6.635V (see earlier discussion) the drop across the (R_2) resistor is 12V – 6.635V = 5.365V. To achieve the required base current requires a resistor value of 5.365V/3.33µA = 1.6MΩ. This will have much less impact on the input resistance than the potential divider bias resistors, but use of a single resistor like this reduces the bias stability – the circuit is more sensitive to variations between individual transistors and changes in conditions such as temperature.

Bootstrapping

Another option is to use bootstrap biasing, which makes use of AC feedback to make a pair of potential divider bias resistors appear to have much higher resistance to the signal, while still retaining acceptably small resistance for DC. A version of the circuit in Fig.7 with bootstrap bias is shown in Fig.13.

The boostrap bias circuit adds a resistor between the bias and potential divider (R_4 in Fig.13), but the input is fed straight to the base as usual via the coupling capacitor (C_1). An additional capacitor (C_3) feeds the output signal back to the potential divider junction. If we assume C_1 is effectively a short circuit for signals, and remembering that the circuit has a gain of close to 1, we find that a voltage equal to input signal is fed to the potential divider junction. The input signal is on the other side of R_4, so the signal voltage across R_4 is very small (it has almost the same signal voltage on each side), which means the signal current through it is very small and its effective resistance is very large (much larger than its actual value). The high effective value of R_3 means that the circuit does not suffer from the relatively low resistance bias components bypassing the high input resistance of the transistor (as happens with the circuit in Fig.7). This effect only applies to signal frequencies, not to the DC bias, which is blocked by C_3. The DC base bias current is small, so there is only a small DC voltage drop across R_3, which can be taken account of in bias calculation if needed.

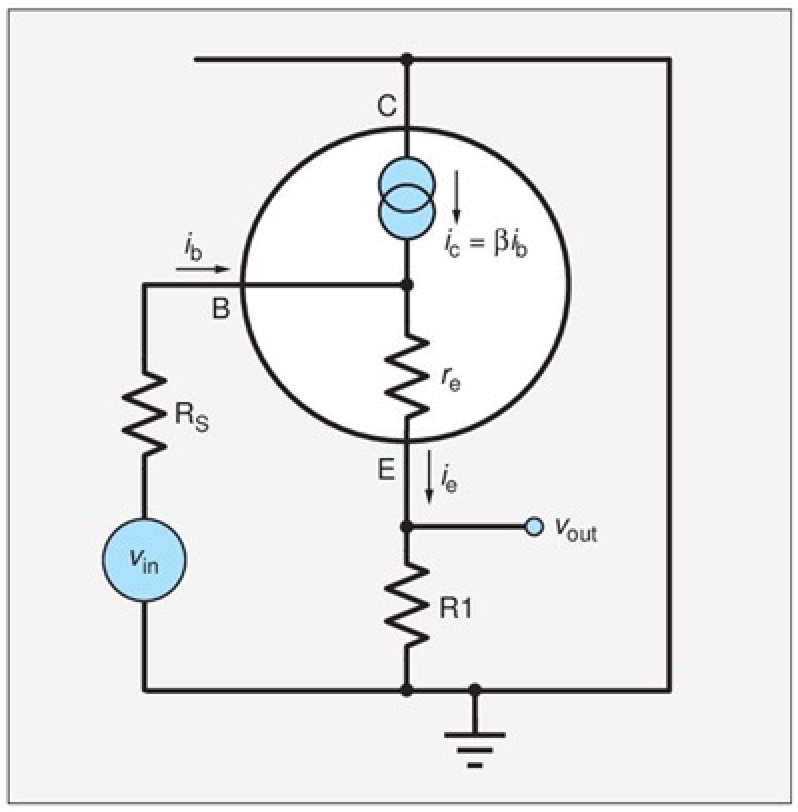


Fig.14. Small-signal model from Fig.5 with source resistance added.

Output resistance

We can calculate the output resistance of the common-collector amplifier by finding the Thévenin equivalent of the small-signal model circuit in Fig.14 (the same as Fig.5, but with source resistance added). The output resistance is the open-circuit output voltage divided by the short-circuit resistance. The open-circuit output voltage is simply v_{out}, which is equal to v_{in} as we have close to unity gain. With the output shorted (short across R_1) we have all of v_{in} across r_e and r_s. If we assume r_e is small compared to r_s (high-impedance source) and ignore r_e, the output short-circuit base current is v_{in}/r_s. Thus the short-circuit output (emitter) current is:

$$i_e = \frac{v_{in}}{r_s}(\beta + 1) \approx \frac{v_{in}}{r_s}\beta$$

Here, we add base and emitter currents and approximate, as we did for the input analysis. Dividing the open-circuit output voltage (using $v_{out} = v_{in}$) by this we get the output resistance:

$$r_{out} = {v_{in}} \Big/ \left(\frac{v_{in}}{r_s}\beta\right) = \frac{r_s}{\beta}$$

The circuit's output resistance is the source resistance divided by the transistor gain – note the symmetry with the input resistance being the load resistance times the gain. Again, there are some approximations here –R_1 and/or the transistors output resistance may need to be included in parallel if they are not significantly larger than r_s/β. On the other hand if r_s is very small r_e will need to be included.

AUDIO OUT

By Jake Rothman

Designing Hi-Fi mini-monitor loudspeakers – Part 2

FD&19MX77

Last month, we introduced the classic British mini-monitor loudspeaker, the LS3/5A. We looked at many of the design, material and component choices and decisions that go into a successful loudspeaker, and we will continue that approach this month, starting with the vital filter at the heart of the electro-acoustic system – the crossover, a circuit designed to steer the right frequencies to the appropriate driver.

Crossover design

Integrating the LS3/5A's two KEF drivers made the BBC crossover filter circuit rather complicated, as shown in Fig.32 and 33. However, the result is an almost seamless integration that is rarely surpassed. The crossover is the most expensive part of the speaker, costing £196.00 for an 'official' (Falcon) pair, using exactly the specified components.

The crossover frequency is 3kHz, which generally sounds good because it is far enough away from the critical 200Hz to 2kHz midrange area and not too high for a 110mm bass unit. It also allows a 19mm dome tweeter to be used, which gives superior dispersion compared to the more normal 25mm size that has to be used for lower crossover frequencies.

This crossover design set the topology for most mini-monitors up until the 1990s, a second-order, low-pass section combined with a mid-range peak equaliser and a third-order high-pass section, as illustrated in Fig.34. More recent speakers aim for a Linkwitz-Riley (LR) fourth-order response because it has a predictable phase result and lends itself to computer design and active systems. Personally, I find the LR crossovers to be more noticeable (ie, less natural) because of their steeper slope and big off-axis power dip. Conversely, there is also a trend for simple first-order crossovers, using a single capacitor and inductor. These are louder and more dynamic on percussive electronic music, but less accurate sounding on vocals.

Loudspeaker drivers are inductive resonant loads, so the response of the filter circuits is not as predictable as with a resistive load. Consequently, the response of crossovers must always be measured

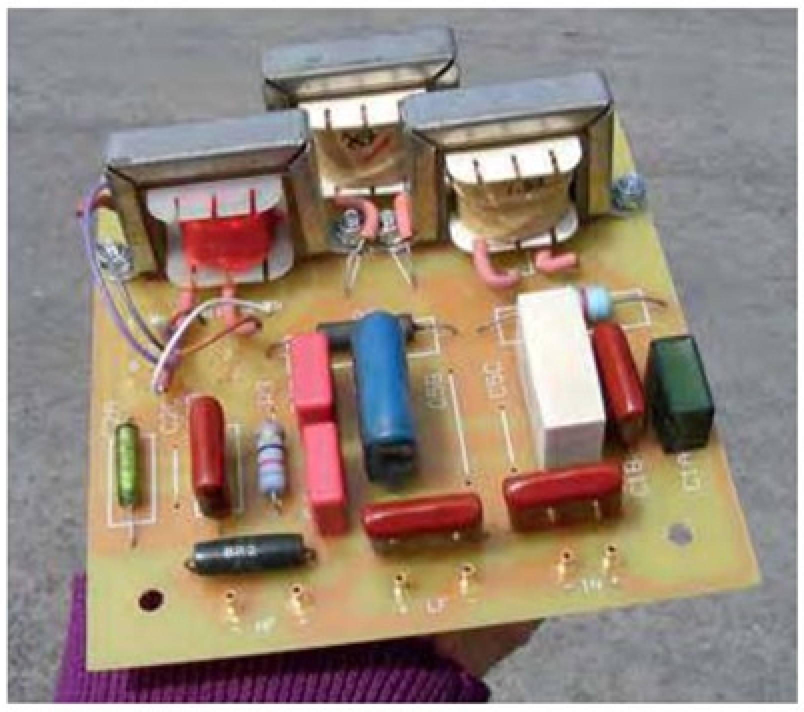
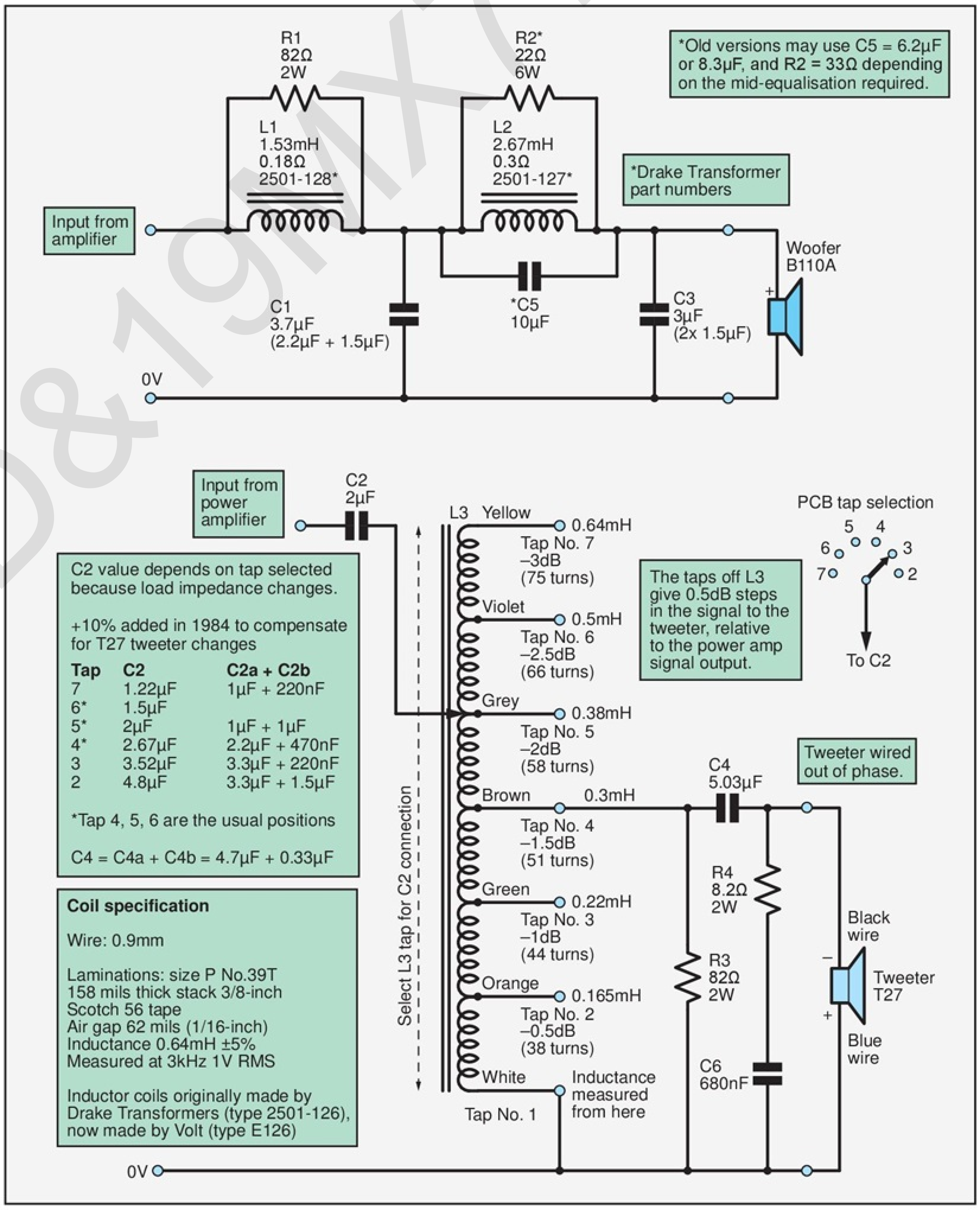


Fig.32. 15Ω LS3/5A crossover assembly.


R1
82Ω
2W
L1
1.53mH
0.18Ω
2501-128*
R2*
22Ω
6W
L2
2.67mH
0.3Ω
2501-127*
Input from amplifier
C1
3.7µF
(2.2µF + 1.5µF)
*C5
10µF
C3
3µF
(2x 1.5µF)
Woofer
B110A
+
0V
Input from power amplifier
C2
2µF
L3
Yellow
0.64mH
Tap No. 7
–3dB
(75 turns)
Violet
0.5mH
Tap No. 6
–2.5dB
(66 turns)
Grey
0.38mH
Tap No. 5
–2dB
(58 turns)
Brown
0.3mH
Tap No. 4
–1.5dB
(51 turns)
Green
0.22mH
Tap No. 3
–1dB
(44 turns)
Orange
0.165mH
Tap No. 2
–0.5dB
(38 turns)
White
Tap No. 1
Inductance measured from here
Select L3 tap for C2 connection
PCB tap selection
5
4
6
3
7
2
To C2
C4
5.03µF
R4
8.2Ω
2W
R3
82Ω
2W
C6
680nF
Black wire
–
Tweeter
T27
+
Blue wire
0V

*Old versions may use C5 = 6.2µF or 8.3µF, and R2 = 33Ω depending on the mid-equalisation required.

*Drake Transformer part numbers

The taps off L3 give 0.5dB steps in the signal to the tweeter, relative to the power amp signal output.

Tweeter wired out of phase.

C2 value depends on tap selected because load impedance changes.

+10% added in 1984 to compensate for T27 tweeter changes

Tap	C2	C2a + C2b
7	1.22µF	1µF + 220nF
6*	1.5µF	
5*	2µF	1µF + 1µF
4*	2.67µF	2.2µF + 470nF
3	3.52µF	3.3µF + 220nF
2	4.8µF	3.3µF + 1.5µF

*Tap 4, 5, 6 are the usual positions

C4 = C4a + C4b = 4.7µF + 0.33µF

Coil specification

Wire: 0.9mm

Laminations: size P No.39T
158 mils thick stack 3/8-inch
Scotch 56 tape
Air gap 62 mils (1/16-inch)
Inductance 0.64mH ±5%
Measured at 3kHz 1V RMS

Inductor coils originally made by Drake Transformers (type 2501-126), now made by Volt (type E126)

Fig.33. The BBC 15Ω LS3/5A crossover (model FL6/23). (top) The low-pass section of the crossover circuit; (bottom) the high-pass section, note the tweeter is connected out of phase relative to the woofer; this is normal in all variants of the LS3/5A.

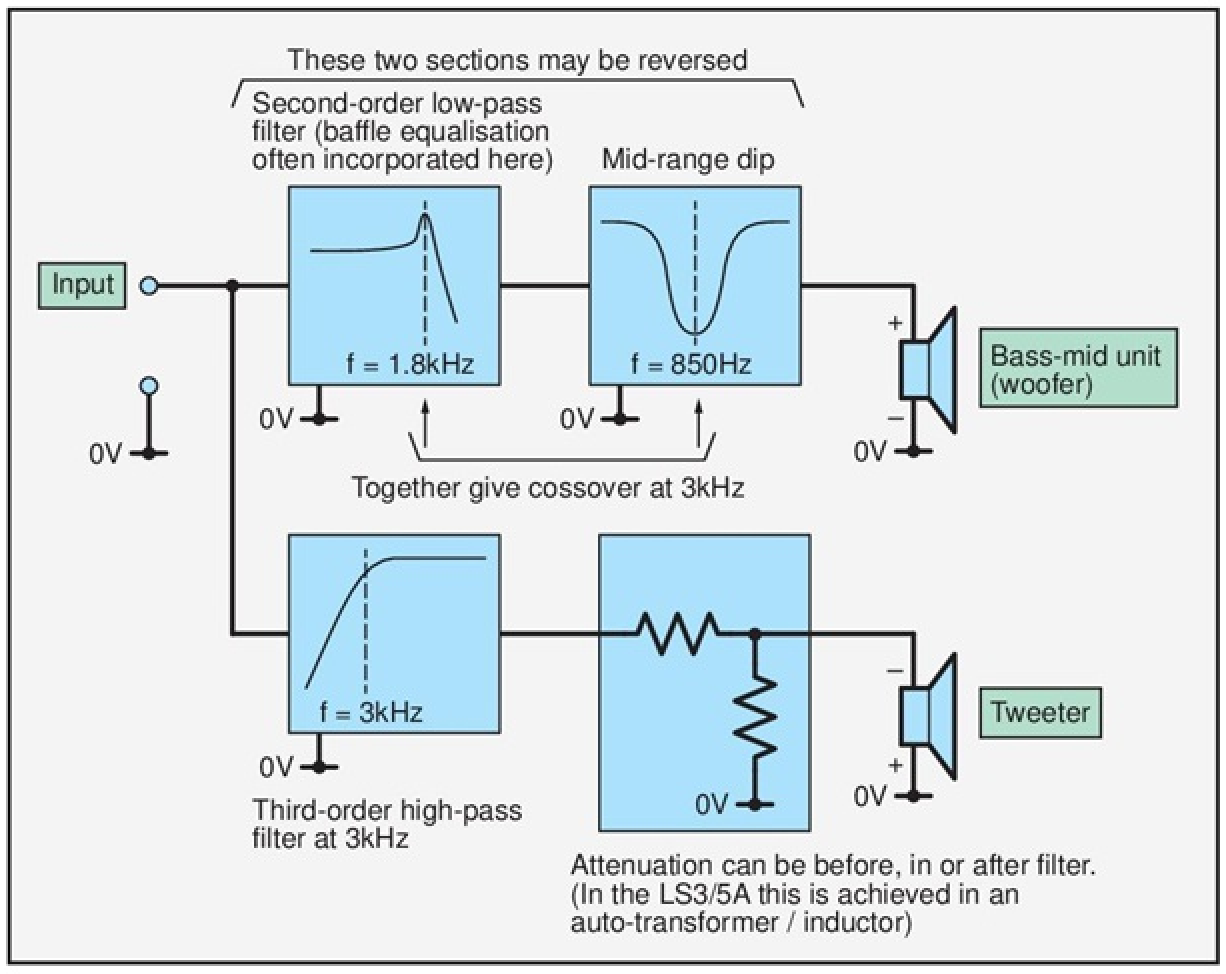

These two sections may be reversed
Second-order low-pass filter (baffle equalisation often incorporated here)
Mid-range dip
Input
f = 1.8kHz
f = 850Hz
0V
0V
0V
0V
Together give cossover at 3kHz
+
–
Bass-mid unit (woofer)
f = 3kHz
0V
Third-order high-pass filter at 3kHz
0V
–
+
0V
Tweeter
Attenuation can be before, in or after filter. (In the LS3/5A this is achieved in an auto-transformer / inductor)

Fig.34. The basic structure of the 15Ω LS3/5A crossover. This circuit structure set the pattern for most high-quality two-way speaker crossovers for 20 years. Note the out-of-phase tweeter.

with the specified drive units connected when plotting the electrical curve. It's also important to remember that the acoustic response has to be added to the electrical response to get the required overall electro-acoustic transfer function. (If a speaker designer says his speaker uses (say) a fourth-order crossover, he means the electroacoustic order, the electrical order of the circuit will be less.)

Another thing to keep in mind is the way the different parts of the electrical curve add up. In the LS3/5A the mid-peak equalisation curve adds to the low-pass filter curve in such a way as to push its effective corner frequency up. Thus, instead of a low-pass filter at 3kHz we require one at 1.8kHz.

Furthermore, in the nightmare that is crossover design, the three dimensional, off-axis response of the system also has to be taken into account. In addition to this, there is the relative phase between the two driver outputs. These factors determine the overall acoustic energy the speaker puts into the room. This makes crossover design an iterative, empirical subject for most engineers, resulting in multiple cycles of: select topology, calculate, tweak circuit, measure, and listen. I still have big development boards stacked with inductors and capacitors from when I did crossover design as a job – see Fig.35.

It is possible to do multiple simulations with software to get 'almost there'. KEF was the first company to do this, and their *KEF Topics* papers are well worth reading (see: **http://bit.ly/pe-oct19-kefus**).

In my early 30s I had good hearing and could knock up a good crossover 'by ear'. Not any more, always beware of 50-year-plus guys designing stuff 'by-ear'! I have measured many a curve that is a mirror-image of their hearing deficiencies. Young women generally have better hearing. Celestion always used them for their swept sinewave rattle test.

I have a pair of the first 'computer designed' speakers, the 1979 KEF 104aB (Fig.36). The 'aB' stands for 'acoustic Butterworth', the resulting electroacoustic curve – a maximally flat 18dB/octave at 3.5kHz response. They still sound amazing, like LS3/5As but 'with bass' because they are big. Off the reference axis, (not directly in front of the speaker as, shown in Fig.37) the curve is not flat, having a loss of energy between 500Hz and 3.5kHz. They have to be in a very dead, almost anechoic room pointing at the listener to sound their best.

Inductors

Getting hold of the original LS3/5A transformer low-distortion Radiometal-cored (50% nickel/iron alloy) inductors is particularly difficult (Fig.38). Volt, the Yorkshire loudspeaker company, now make these inductors exclusively for Falcon. I've resorted to winding my own, not as difficult as you would expect if you have an inductance meter – an essential piece of kit for any crossover maker – and can obtain Belclare or Drake cores from old BBC engineers' junk boxes (although this is not an option for most people). These cores give relatively lower third-harmonic distortion compared to normal iron-dust cores (Fig.39).

Air-core coils (Fig.40) give the lowest distortion, but have higher resistance, are much bigger and cost twice as much as ferrite cored types simply because much more copper is needed. A ferromagnetic

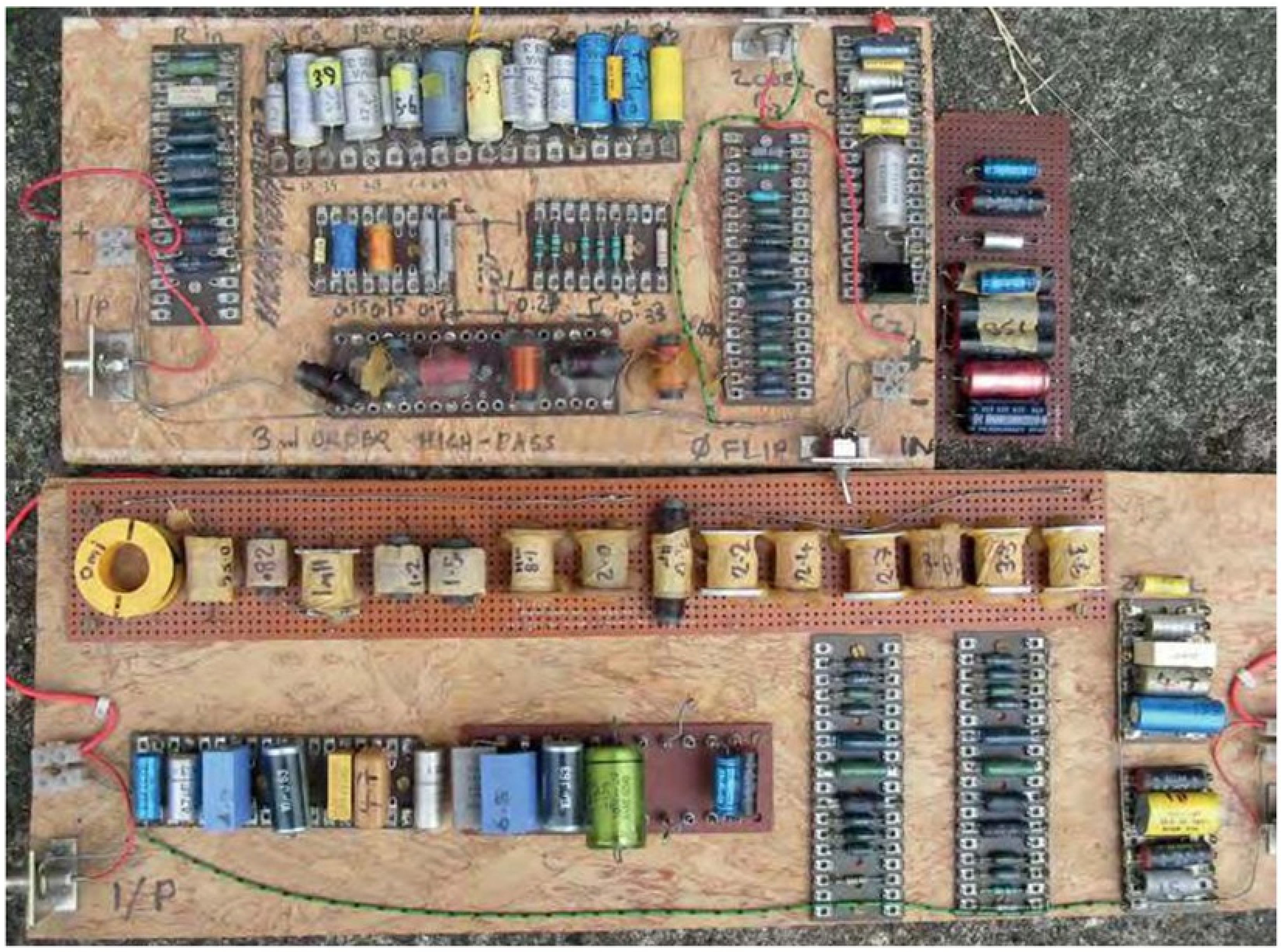


Fig.35. One of my crossover development boards from the 1990s. Since most speakers have a low impedance and only a small range of crossover frequencies, a great range of values is not needed. Here, I can use: 0.1 to 6mH for the inductors, 0.22µF to 22µF for the capacitors and 0.22 to 330Ω for the resistors.

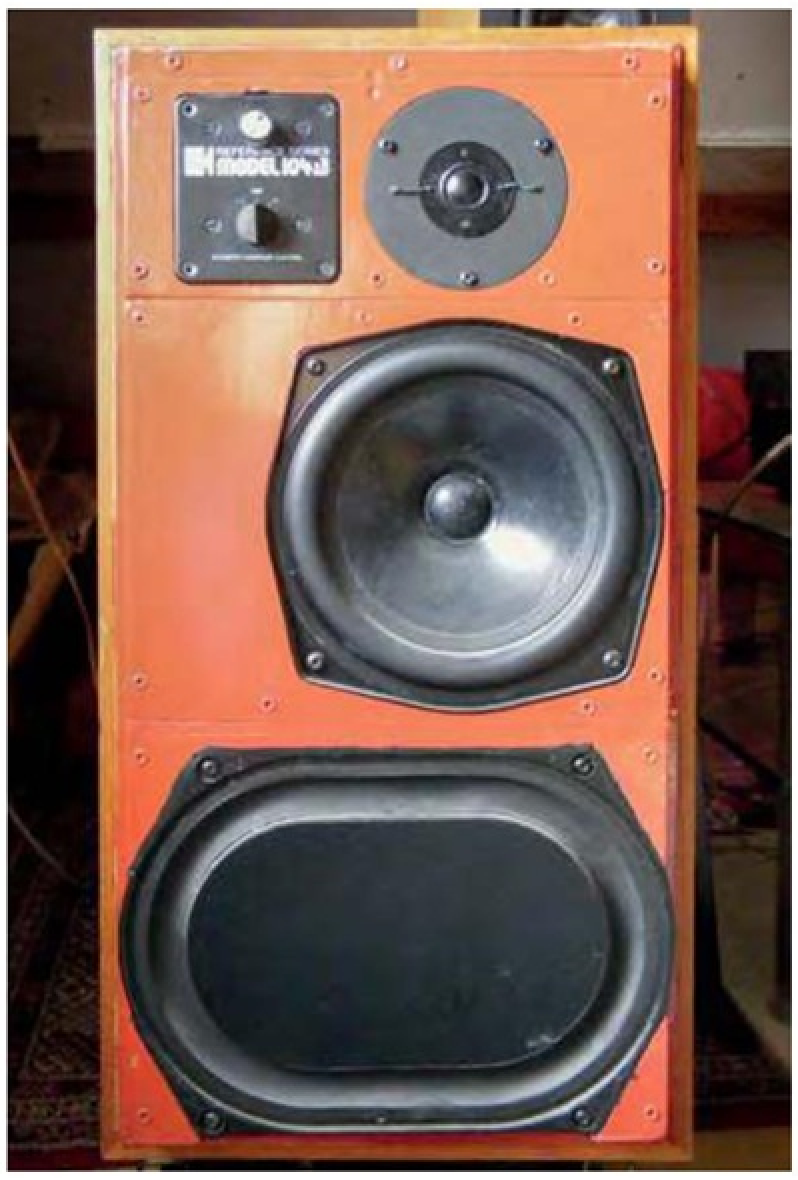


Fig.36. The KEF 104aB is one of the nicest sounding speakers because of it's smooth slightly recessed midrange. It also has lower distortion due to its short voice-coil in a long gap.

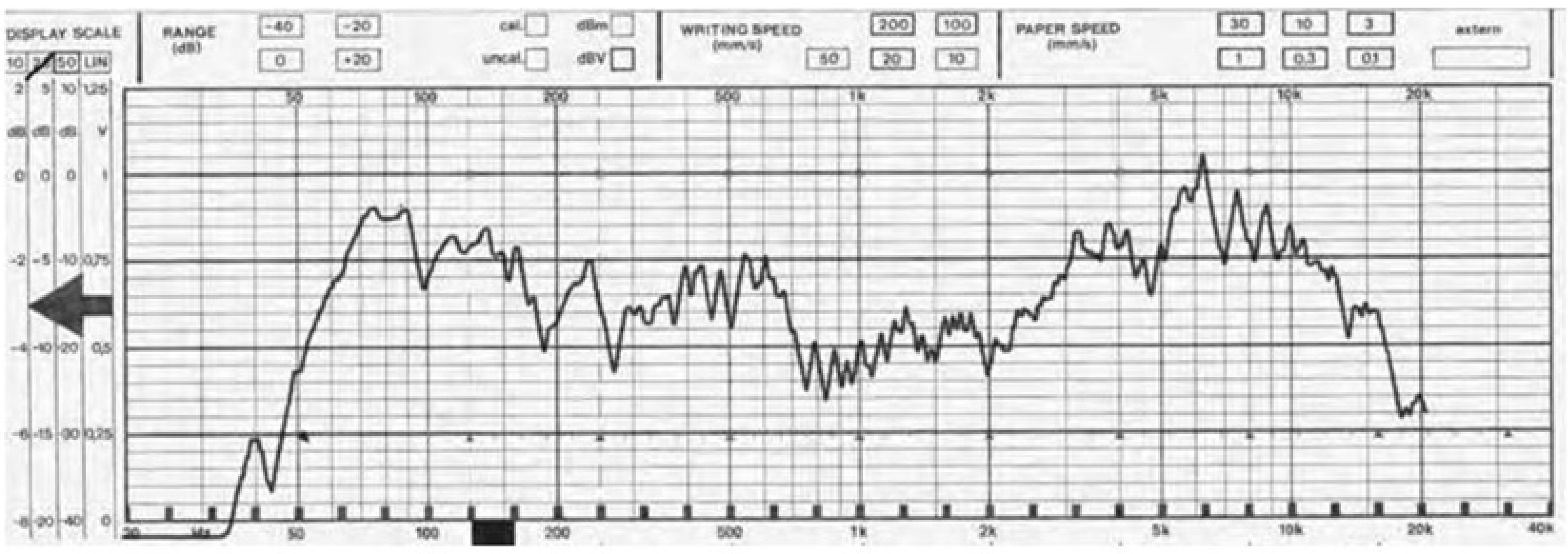


Fig.37. Frequency response of 104aB.The fundamental resonance of the tweeter was suppressed using a notch in the crossover.

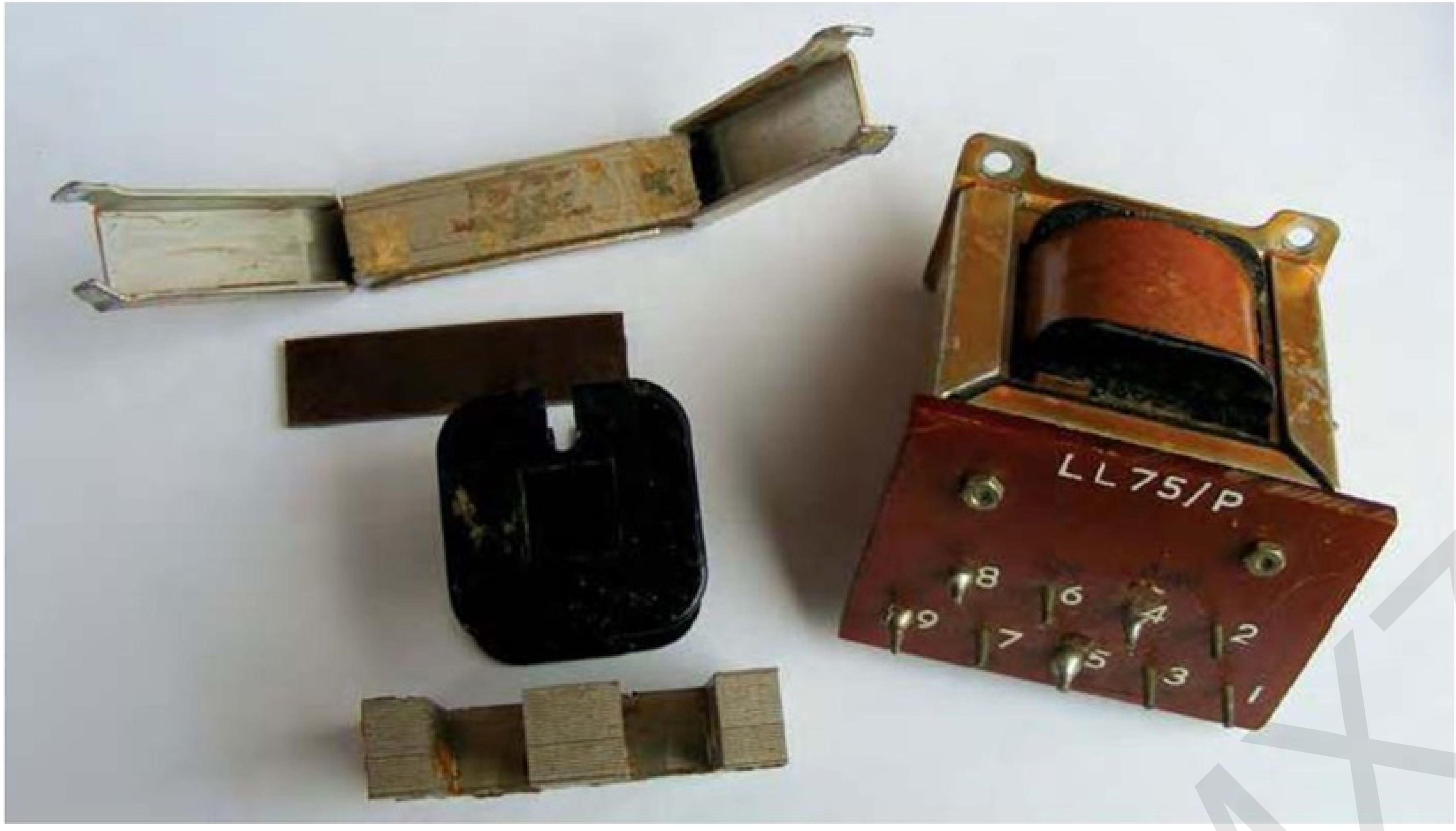


Fig.38. A Radiometal loudspeaker inductor used by the BBC. It combines the low distortion of air-cored inductors with the low resistance of ferrite cores. Nickel-based laminations have a bright appearance rather than the dull grey look of silicon-iron transformer cores.

core reduces the turns of wire needed for a given inductance because it has a magnetic-field-boosting parameter called 'permeability'. It is easier to wind air core coils yourself since any non-magnetic non-conducting former can be used, and we'll wind some later.

LS3/5A crossover PCBs available on eBay, usually from Hong Kong where there is a thriving 'unofficial' (pirate) LS3/5A industry. A new Chinese original BBC design 15Ω example is shown in Fig.41 for the classic BBC circuit shown in Fig.33. Note that the non-standard capacitor values, such as C1 (3.7µF), are made up from parallel combinations of standard E6 series values. For C1, a 2.2µF and a 1.5µF are used (denoted C1a and C1b) respectively. Likewise, C2 (3µF), is made of two 1.5µF capacitors in parallel. The capacitors are plastic-film (originally polycarbonate) with 5% tolerance and a voltage rating of 63V minimum.

A board for KEF's 11Ω version is also available (shown in Fig.42a), which is particularly useful for most standard configuration mini-monitor crossovers. A populated 11Ω KEF-made board is shown in Fig.42b. Falcon also do a similar PCB called the '23aB' (shown in Fig.43a), which is designed for the 23aB circuit shown in Fig.51.

It can be argued that because of the size of crossover components, a PCB isn't really needed at all. Hardwiring, 1930s style, using M4 solder-tags screwed into wood (shown in Fig.44) is superior, having lower resistance and higher mechanical strength. Inductor orientation is important to avoid mutual coupling. Keep them as far apart as possible and at right angles to one another, using non-magnetic fixings – do remember that metal boxes and steel screws can cause inductance changes and distortion. The Radiometal inductors in the original LS3/5A are not so critical in this respect, having a more enclosed magnetic field. Hot glue is often used to stop the components rattling.

Active crossovers

Passive crossovers are expensive because they use large coils and non-polarised capacitors which are rare, having almost disappeared from mainstream electronics in the last 30 years. The coils are typically £3.00 to £12.00 each and the capacitors £1.00 to £5.00. Even the 3W-rated resistors cost over £0.50 each. The best solution is to go active and put the filter before two power amplifiers, one each for the woofer and tweeter.

This involves screwing a board containing a couple of power amps and a low-level active filter to the back of the speaker cabinet. It enables standard small passive components, op amps and lower-power power-amps to be used, reducing the total system cost and distortion.

Active speakers have not been as popular as they should be because the potential cost reductions have not been passed on to the customer. I've never understood why an active monitor often costs twice as much as the passive version. I suspect it's production logistics, insurance and labour costs. One aspect that has also been ignored is that passive loudspeakers last 30 years while the active electronic part usually lasts about ten, and this has also put people off. I've had active speakers in the workshop with popped SMPS electrolytics, blown Class-D power-amp chips and burnt-through SMT boards that have only just lasted beyond the guarantee. Other problems are no circuit diagrams and test data. It's a great time to be in the landfill business! The only speakers I've seen where the electronics is of commensurate quality were in the Meridian M2 design.

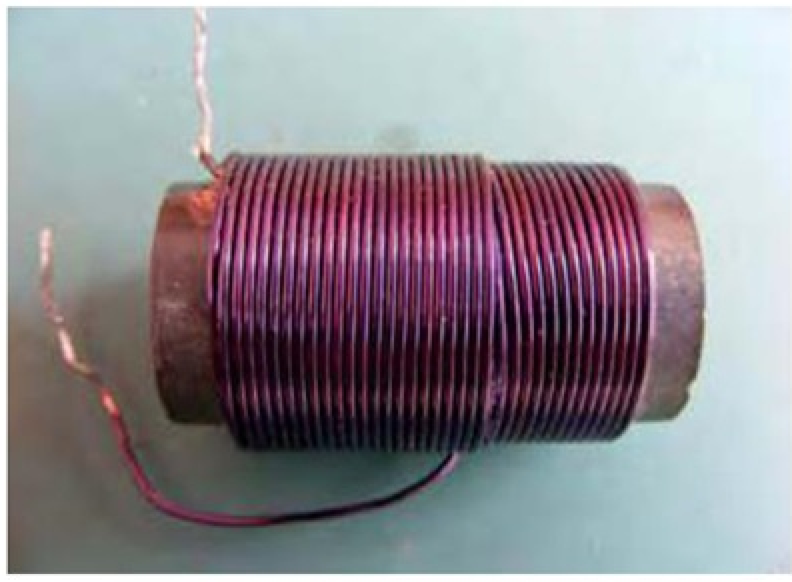


Fig.39. A ferrite-cored inductor – Malcolm Jones of Falcon invented a trick where the turns of wire were bonded together by passing a large current though it. This was sufficient to just soften the insulation enough to get it to stick together without shorting. KEF and Celestion used to battle with glue and tape for their inductors.

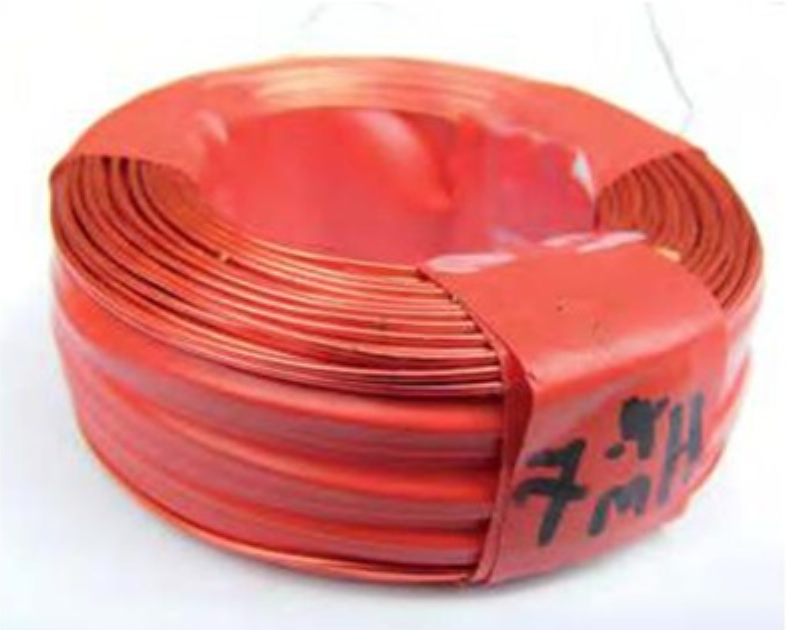
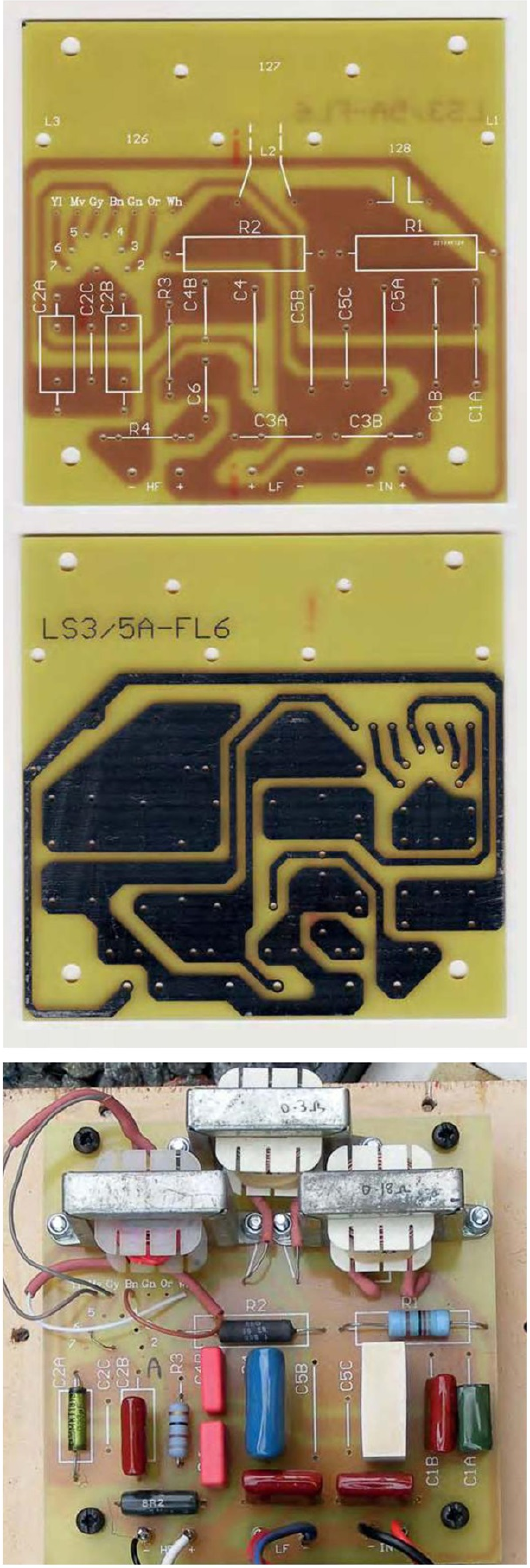


Fig.40. An air-cored coil – these need loads of turns to get the required inductance which pushes the cost and resistance up. They have very low distortion and are easy to make one's self as this example shows. It's 'held together' with red insulating tape.


LS3/5A-FL6
L1
L2
L3
R1
R2
R3
R4
C2A
C2C
C2B
C4B
C4
C5B
C5C
C5A
C6
C3A
C3B
C1B
C1A
Yl Mv Gy Bn Gn Or Wh
HF
LF
IN
0·3Ω
0·18Ω

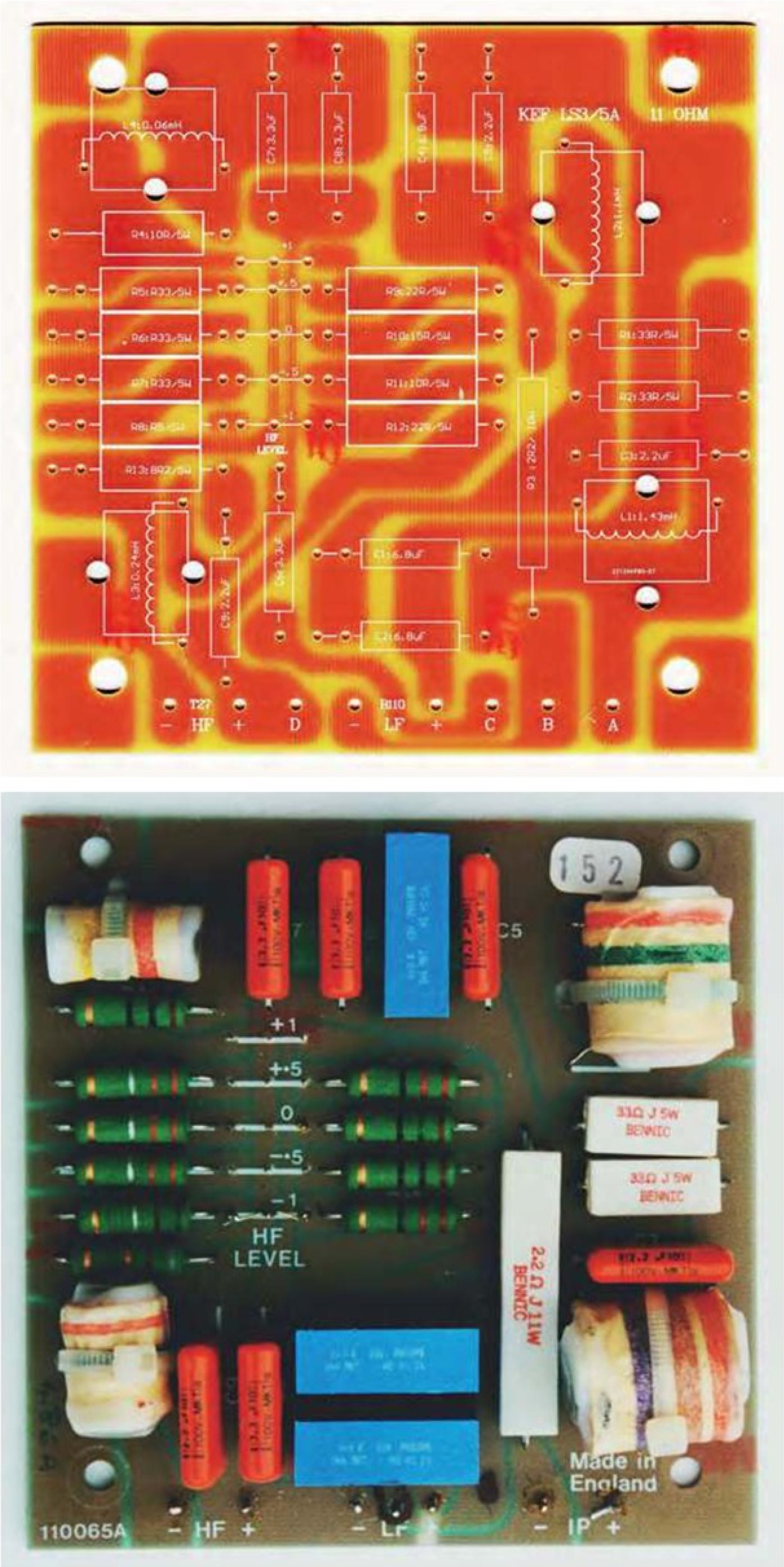
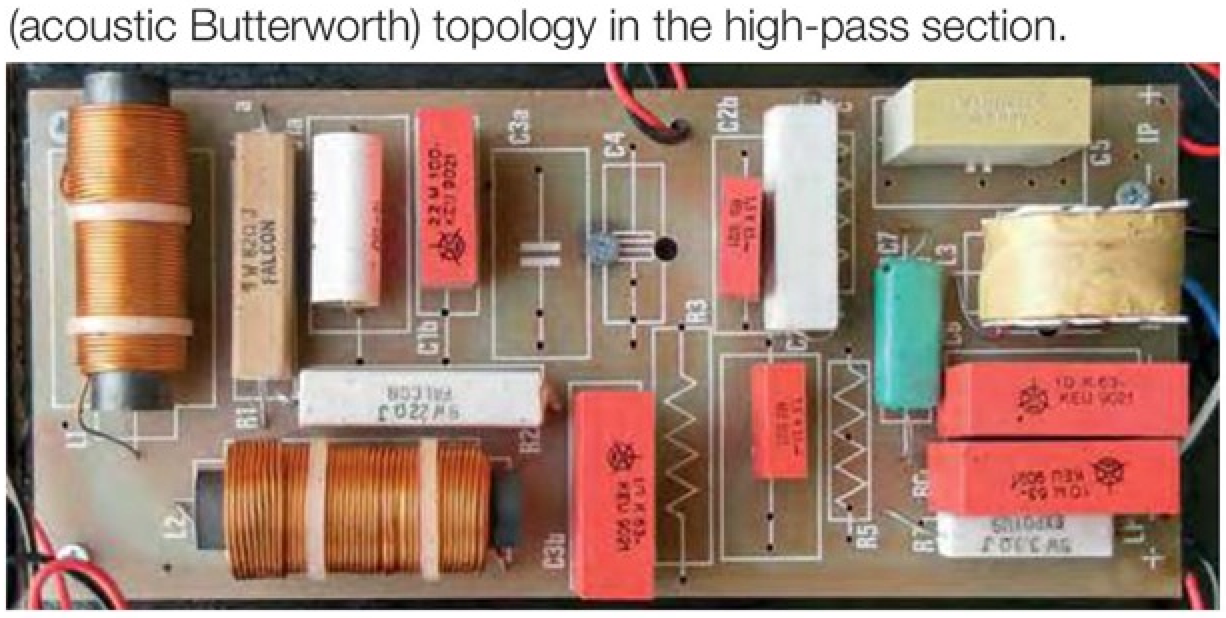

KEF LS3/5A 11 OHM
HF LEVEL
+1
+.5
0
−.5
−1
HF
D
LF
C
B
A
152
Made in England
110065A
IP

Fig.41. (top left) LS3/5A PCBs can be bought on eBay from Hong Kong (item 191278013041). (Technically, this is IP theft, but I suspect the BBC is not interested in chasing PCB manufacturers); (bottom left) Close up of Fig.32 showing the board populated.

Fig.42. (top right) A KEF-designed 11Ω crossover board from eBay (item 201203704788). (below) a populated version of this board. This is ideal for building standard ferrite inductor designs of the topology described in Fig.34.

Fig.43. (below) Falcon 23aB crossover board layout, with BBC 15Ω low-pass section using ferrite inductors and the 104aB (acoustic Butterworth) topology in the high-pass section.



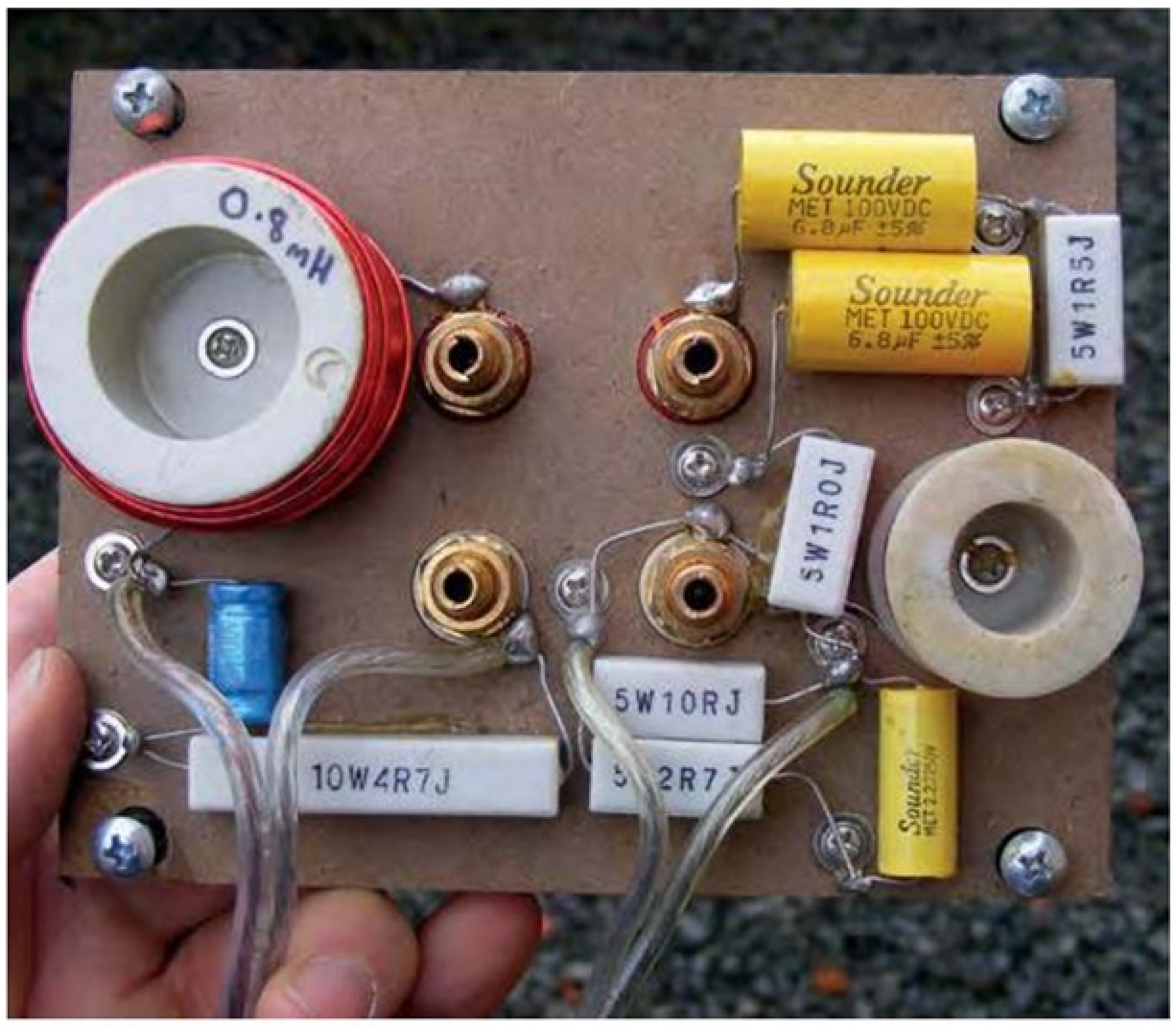


Fig.44. (left) In passive crossovers, the components are so large with impedances so low that 1930s-style building on wooden boards gives excellent results – but with high labour costs (fortunately, irrelevant for home constructors).

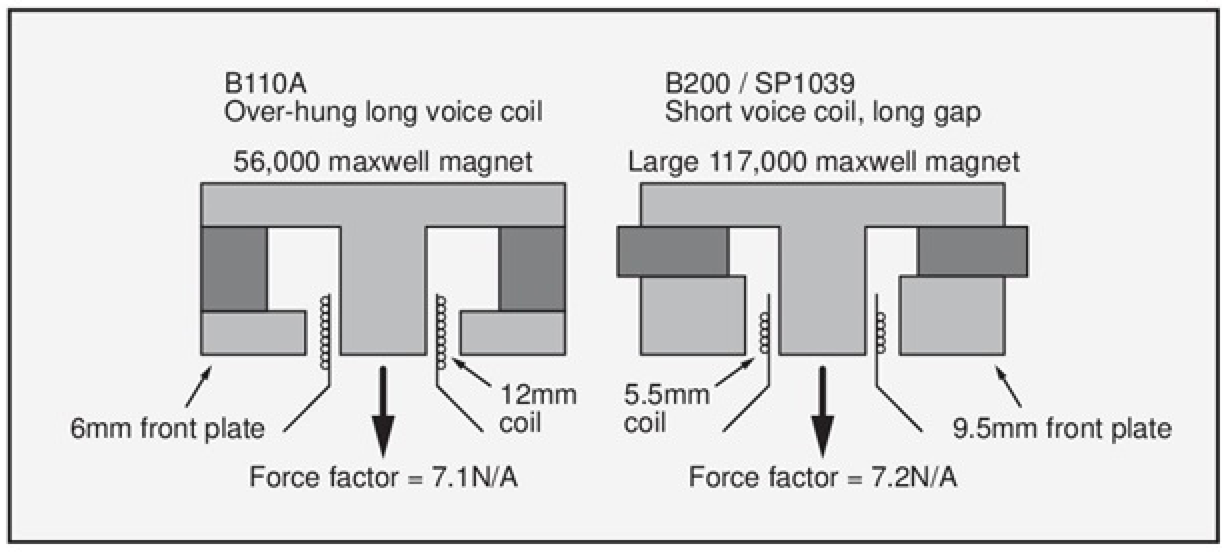


Fig.45. (above) Coil geometries giving large linear excursions: (left) long-coil, short-gap used in the B110A, max linear excursion (x_{max} 6mm peak-to-peak); (right) short-coil, long-gap used in the 104aB (x_{max} 5mm peak-to-peak); and ATC bass drivers.

Limitations

There are limitations with small speakers governed by the immutable laws of physics. They will not go loud or give deep bass. However, the mid and top response can be as good if not better than larger speakers. Significant bass below about 75Hz is generally impossible, which is a realistic lowest fundamental resonance for a small sealed enclosure of five litres (0.2 cubic feet). To even get the bass down to this point efficiency has to be greatly sacrificed and to bring what bass there is into balance means attenuating the mid and top by about 10dB. Also, because the cone area is so small the cone has to travel much further. This means using a very long voice coil to get what is called a 'long-throw driver'. A long coil is only partly immersed in the magnetic field, which means two thirds of the coil is doing nothing (Fig.45, left). The system can be reversed, having a short coil in a long gap (Fig.45, right), but then two thirds of the magnet is doing nothing.

Although giving superior linearity and reduced coil mass, the short coil results in a very expensive loudspeaker, the magnet assembly being the most costly part. Mini-monitor bass units often have a magnet that is almost the same diameter as the cone. Efficiency is also reduced by measures used to reduce colouration. Typically mini-monitors have an efficiency of 82 decibels per watt (dB/W) at one metre, but a 5-inch bass unit can only handle around 40W. These factors limit the maximum output to typically 96dB at around one metre for a stereo pair. This is loud enough for me with my ageing ears that were once very sensitive, but for many young high-volume fans it is not enough. If you like gut-thumping bass, go for a large modern bass-reflex design with a paper cone. If you listen to small-scale acoustic music, such as soft jazz and choral, and appreciate stereo depth and low tonal colouration, then mini-monitors are ideal.

There is normally an optimum internal air volume for a particular bass driver. For the B110A it is around five litres. This is quite small because the B110 has a very floppy cone assembly, ie only a small amount of force is needed to move it a long way. This is called 'compliance' and is normally expressed in meters per newton (m/N). More modern drive units generally have stiffer (less compliant) suspensions and higher resonant frequencies. These need a bit more internal volume, say 7.5 litres, possibly with a small reflex port, or the resonance will end up nearer 85Hz.

There are now many online calculators for speaker enclosure formulae. An advantage of small cabinets is that they emit less acoustic coloured sound from their side and back panels compared to larger boxes, because of the reduced surface area. Their narrow fronts also give more natural dispersion on voices, possibly because they have similar dimensions to a human head. One of the reasons small speakers seem to work well in typically small English Victorian living rooms is the bass boost that occurs due to the proximity of the heavy masonry and lath-and-plaster walls. This effect is called 'boundary reinforcement'. If you live in a modern open-plan/timber-framed house then there will be little of this bass boost available. Another factor is midrange absorption from carpets and soft furnishings.

Distortion

Loudspeakers often have very high distortion figures compared to amplifiers and digital recording media. The lowest total harmonic distortion that can be achieved is 0.1% in the midrange. In the bass end it's much higher, typically 1% to 3% mainly second harmonic. Along with the third harmonic distortion from ferrite inductors and some capacitors in passive crossovers, the situation can seem worrying. However, this type of low-order harmonic distortion which increases with level is unnoticeable to human ears, (on simple music). It can be enhancing and electronic music producers often add it to the mix via FET limiters and other processors.

As an illustration of speaker electro-mechanical non-linearity, Fig.46 shows a graph (the only one I have seen) showing the cone excursion for a Richard Allen HP8B bass unit, an excellent eight-inch woofer from the late 1970s. Despite having a 15mm-long voice coil, the linear bit of the curve is only about 4mm. It is also asymmetrical, these factors account for the copious amounts of distortion most speakers produce. The Wavecor bass drivers from China and ATC drivers are among the few modern designs that address this problem.

Impedance

The original LS3/5A has a nominal impedance of 15Ω (see Fig.47), which means the amplifier driving it gives reduced power. This does minimise solid-state amplifier distortion at low powers, however; and I suspect this

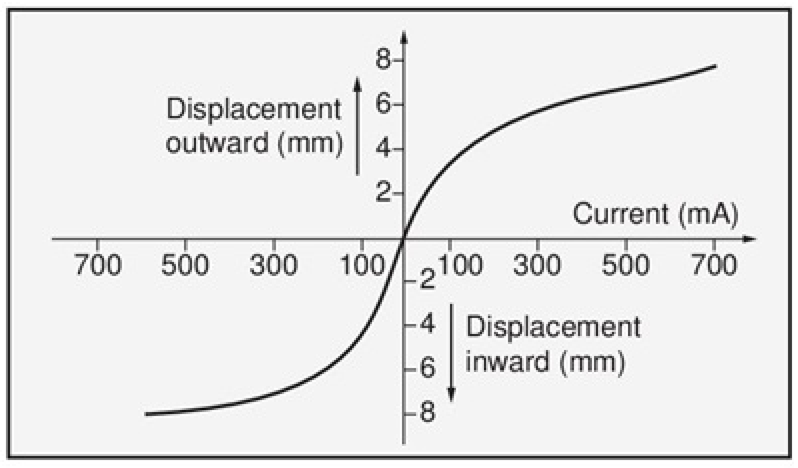


Fig.46. Displacement graph for the Richard Allen HP8B bass-mid driver – note the asymmetry and non-linearity.

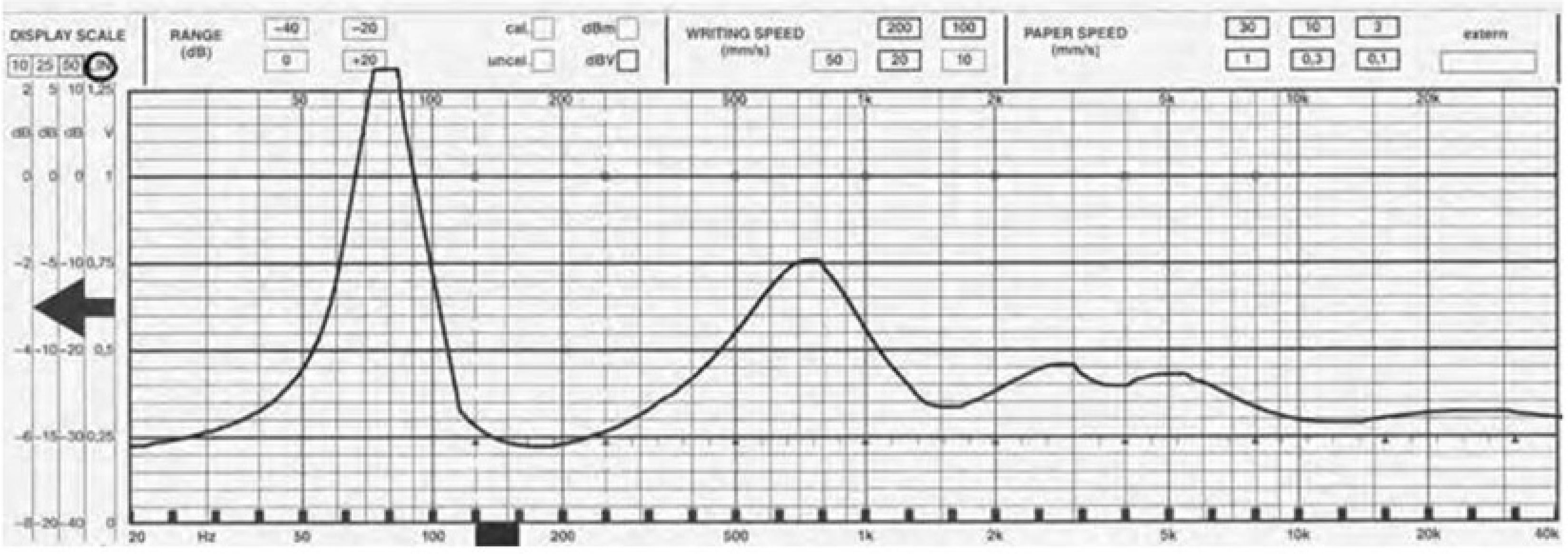


Fig.47. Impedance curve of the LS3/5A – note the bass resonance at 80Hz. This is a bit out of spec, it should be 75Hz. (The impedance plots still have a log frequency scale, but the impedance on the vertical axis is linear. In this case it is 2Ω per division, or 50Ω for the whole scale).

may contribute to the clarity of the LS3/5A in many Hi-Fi systems. Some speaker designers attempt to gain extra power by dropping the impedance to 4Ω, such as the Harbeth HPL3, which dips to 3.5Ω at 50Hz. This strategy fails because the real limiting factor in acoustic power output is the useable cone excursion and surface area. It also makes standard transistor power amps generate around four times more crossover and power-supply-induced distortion. It's best to stick with standard 8Ω drivers for this reason if doing one's own design. Also, 4Ω drivers are less efficient, since they fit less copper into the available space in the voice-coil.

Amplifier

When choosing an amplifier for the LS3/5A, it should be a type that can swing a lot of volts, at least $50V_{pk\text{-}pk}$. Massive current delivery into a low impedance is not necessary. Lateral MOSFET amplifiers with their high-voltage ±50V rails, such as the classic Maplin 50W MOSFET and Hitachi designs fit into this category. It would be worth using the MX50 board (see May 2017) with the Exicon MOSFETs, which I will do sometime. Lateral MOSFETs are also ideal because they are the most reliable audio output power devices. I'm a great believer in using a £200 amplifier and digital player with £2000 loudspeakers because the speaker is the weakest link. This is the audio engineer's approach, Hi-Fi people will definitely disagree!

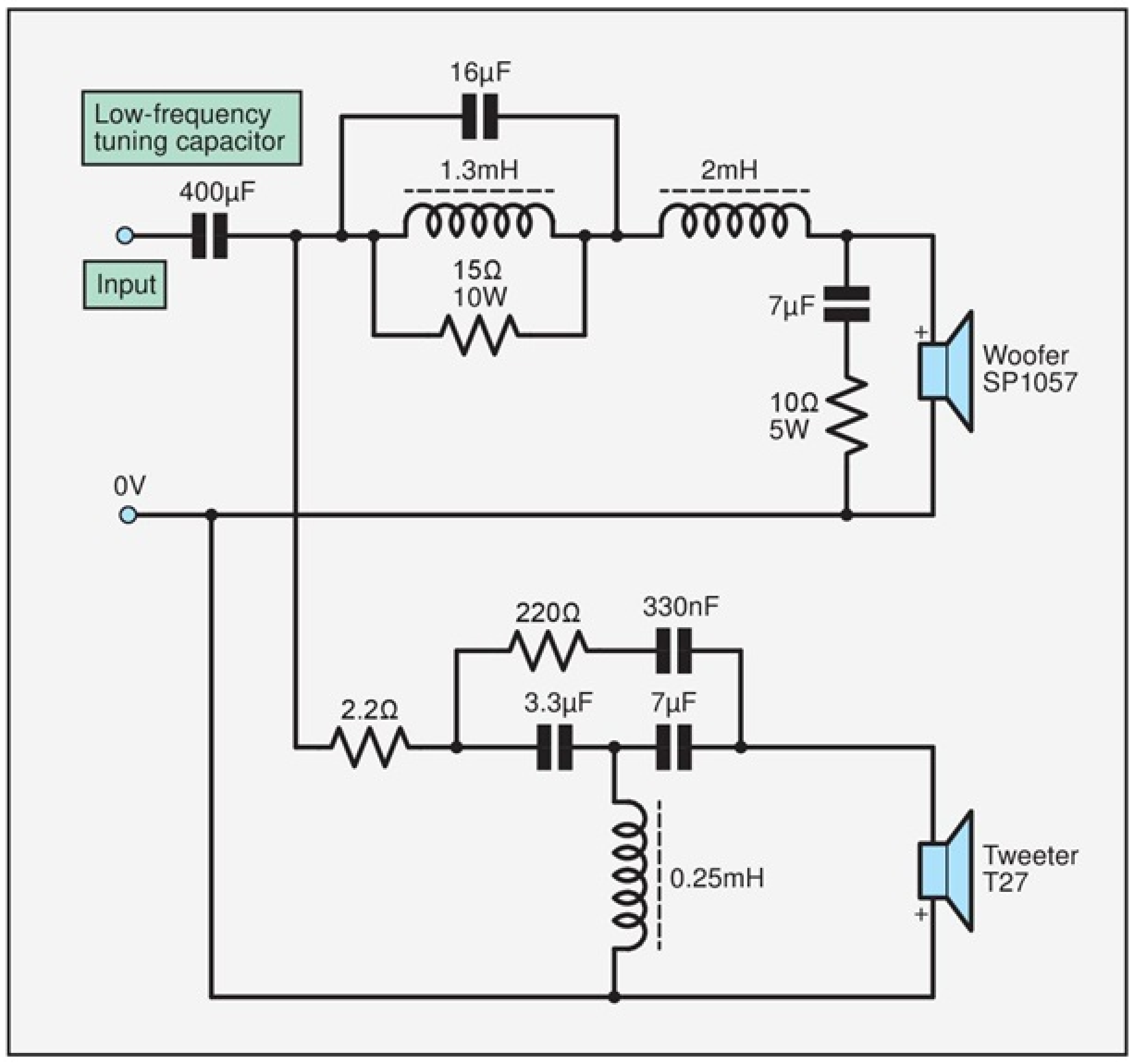


Fig.48. Kef R101 crossover circuit (simplified with the protection circuitry omitted).

LS3/5A variants

KEF made their own speaker similar to the LS3/5A called the R101 which had a much flatter frequency response. It used an 'improved' version of the B110A called the B110B which had a PVC rather than rubber surround. This damped the cones' break-up modes better, but spoilt the bass transient response due to mechanical hysteresis. Subjectively, it sounded more coloured than the BBC design, despite its much smoother curve. A simplified version of the crossover omitting the protection circuitry is shown in Fig.48. However, the BBC used live vs reproduced sound comparisons and found that for small speakers a flat curve was not optimum; there had to be peaks and dips in the right places for it to 'sound' flat.

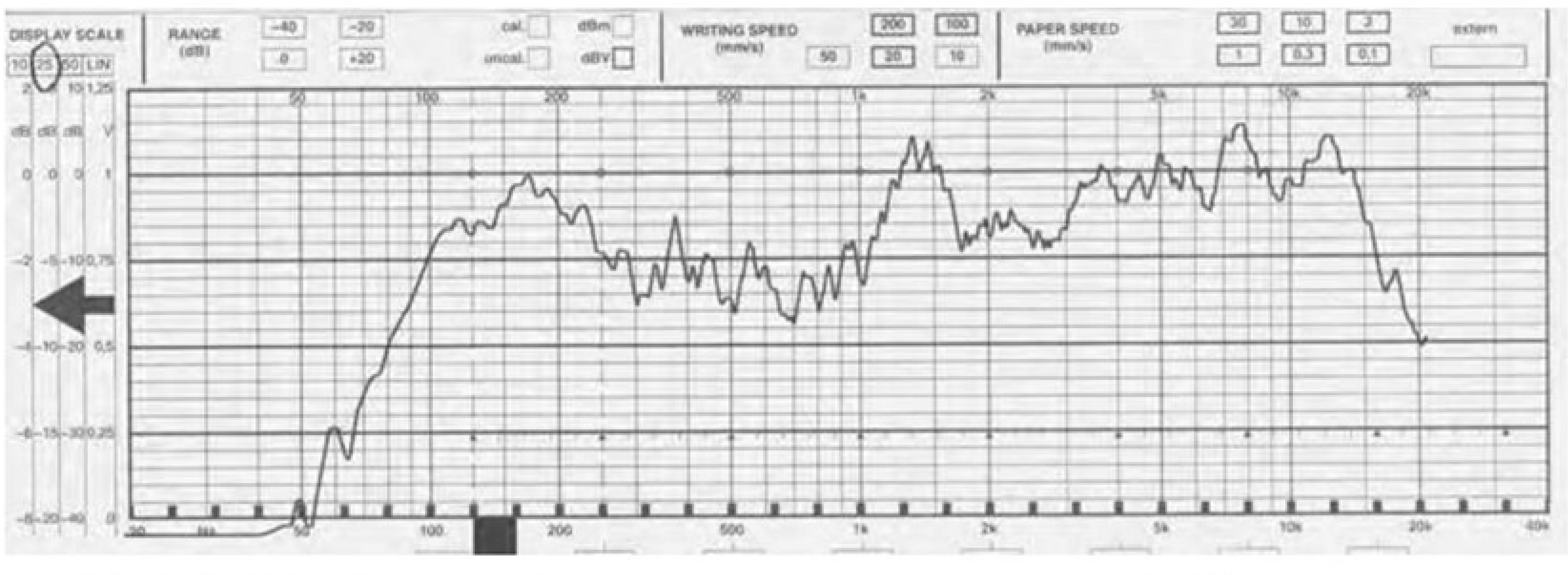


Fig.49. Typical LS3/5A *acoustic* frequency response – measured with a microphone.

A typical LS3/5A curve is shown in Fig.49. It can be seen there are significant peaks at 150Hz partly due to the high (1.2) Q of the bass resonance. It displays a 1.2kHz cone resonance typical of Bextrene cones with a rubber surround, sometimes described as a 'quack colouration', and an 8kHz to 10kHz tweeter peak. The mid peak became more troublesome to control with later versions of the B110A when flame retardants were added to the Bextrene. Because of this, KEF redesigned the whole LS3/5A for the BBC in 1988, making a new version of the bass unit (B110C) with a high damping coefficient PVC surround and ferrite inductors. This had an 11Ω average system impedance compared to the original 15Ω BBC version. The circuit is shown in Fig.50 for reference. Again, this just did not sound as good as the original, having higher distortion and poorer bass, which is why modern LS3/5A manufacturer Falcon are only making the original version. As a final note, however, the variants can work well if relieved of their bass burden by use of an active subwoofer.

Other versions occasionally encountered are the early kits sold by Falcon. They used a crossover called the Falcon 23aB which used the original BBC low-pass section with the KEF 104aB T27 tweeter section. This 'new' high-pass circuit is shown in Fig.51 (and also Fig.43). A 104aB crossover assembly is shown in Fig.52. I think this high-pass filter sounds better than the proper LS3/5A design because it suppresses the tweeter's fundamental resonance.

KEF also did a 'Home Constructor Series' mini-monitor called the CS1A, which was designed to mimic the LS3/5A and used the same bridged-T circuit trick in Fig.51. The CS1A crossover circuit is shown in Fig.53. There was also a popular Wilmslow Audio kit from the 1980s called the Micro Monitor that was based on this. The electrical crossover curve of the proper LS3/5A is shown in Fig.54.

Finally, the JR149 (Fig.3) crossover had a unique layout with a circular PCB mounted in the base, as shown in Fig.55.

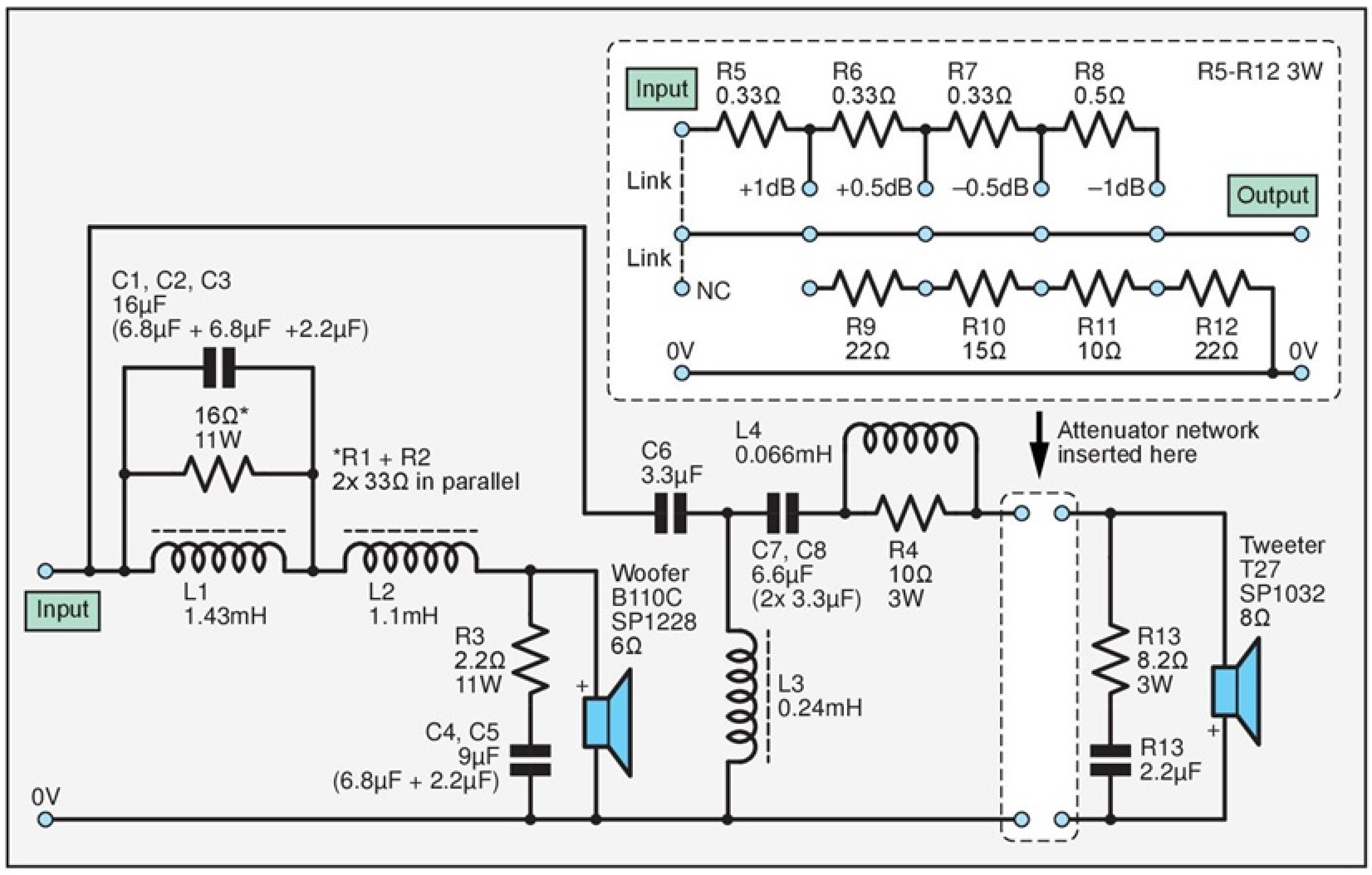

Input
R5
0.33Ω
R6
0.33Ω
R7
0.33Ω
R8
0.5Ω
R5-R12 3W
Link
+1dB
+0.5dB
–0.5dB
–1dB
Output
Link
NC
R9
22Ω
R10
15Ω
R11
10Ω
R12
22Ω
0V
0V
C1, C2, C3
16µF
(6.8µF + 6.8µF +2.2µF)
16Ω*
11W
*R1 + R2
2x 33Ω in parallel
C6
3.3µF
L4
0.066mH
Attenuator network
inserted here
Input
L1
1.43mH
L2
1.1mH
R3
2.2Ω
11W
C4, C5
9µF
(6.8µF + 2.2µF)
Woofer
B110C
SP1228
6Ω
C7, C8
6.6µF
(2x 3.3µF)
R4
10Ω
3W
L3
0.24mH
R13
8.2Ω
3W
R13
2.2µF
Tweeter
T27
SP1032
8Ω
0V

Fig.50. The 'new' post-1987 KEF-designed 11Ω LS3/5A crossover circuit.

It seems it's virtually impossible to improve the LS3/5A, there is some overall synergy about the design that is difficult to quantify, which is why Falcon are making an exact copy of the 1970s design.

There was a 'crazy' speaker that used the same sized cabinet and B110 bass unit called the 'Kan', made by that British Hi-Fi leviathan, Linn. It was quite popular in the 1980s, even the editor had a pair. (In fact, they still have quite a following.) This was not designed to be accurate, but entertaining with Linn vinyl sources. It had no mid-range flattening, sounding very punchy and loud. It had to be used against a wall to provide a compensating bass boost. This position destroyed the illusion of stereo depth, which requires stand-mounting away from walls. These speakers did not sound so good with CD sources, but I have used a parametric equaliser with them to good effect.

PE 'LS3/5A style' monitor

If you are rich, then do buy a new pair of 1975-designed LS3/5As. If not, I have arranged with Jerry Bloomfield, the MD of Falcon, to supply parts for a PE 'LS3/5A style' monitor. Unlike most imitators, it will use the most important part, the B110A bass unit, which imparts most of the LS3/5A 'character'. Descending into the swamp of Hi-Fi subjectivism, this is mainly due to the Bextrene 'quack' which is enhancing. All cone materials have some colouration; there is the 'polypropylene sound', which is lower in frequency and Q, but dulls rather than enhances. Modern bass units do not sound as good because of their higher resonance, no doping and higher surround damping. They will go louder, however – essential for all fans of Spinal-Tap-style approaches to volume (**https://youtu.be/s9F5fhJQo34**).

New theory

Another factor, which I do not yet fully understand, is the poor sound quality that comes from using aluminium voice-coil formers (Fig.56), which are now the norm for raised power handling. The B110A uses Nomex, a polyamide 'paper' which is an electrical insulator, but it tends to blister when it gets hot in conditions of high humidity, causing voice-coil rubbing. It is usable up to 250°C, but does sound very good. Aluminium is a good conductor of heat and is usable up to 350°C. It is also a good conductor of electricity, which causes a damping problem. This is what causes the problem. We have all seen the YouTube video of dropping a magnet down a copper pipe (**https://youtu.be/lnY9u8TfFKI**). It shows how the magnet is slowed due to currents induced in the pipe, which causes an opposing magnetic force. It's a prop I use as one of my classroom demos. This slowing-damping effect also occurs with aluminium voice-coil formers and can readily be felt when moving a detached cone/coil assembly up and down in the magnetic gap. These voice coils normally have a gap to avoid creating a complete shorted turn to reduce this effect, but it doesn't really work, possibly because of eddy currents in the aluminium.

In addition to the damping problem, there is another, more insidious effect, which is increased intermodulation distortion in the mid-range, which is clearly audible, although I've not measured it yet. This may arise from the aluminium eddy currents being coupled by transformer action into the iron pole-pieces of the speaker magnet assembly. This iron has high hysteresis, which when an induced current flows through it will, in turn, couple a distorted current into the voice-coil current. ATC have got round this problem (which occurs to some extent in all electromagnetic transducers) by using electrically insulating sintered-iron rings on the centre pole and front plate. Another trick, used by Seas and Wavecor, is to put copper caps and shorting rings on the centre pole. The lower resistance of the copper shunts away any currents from the higher resistance iron. This loose-coupling distortion

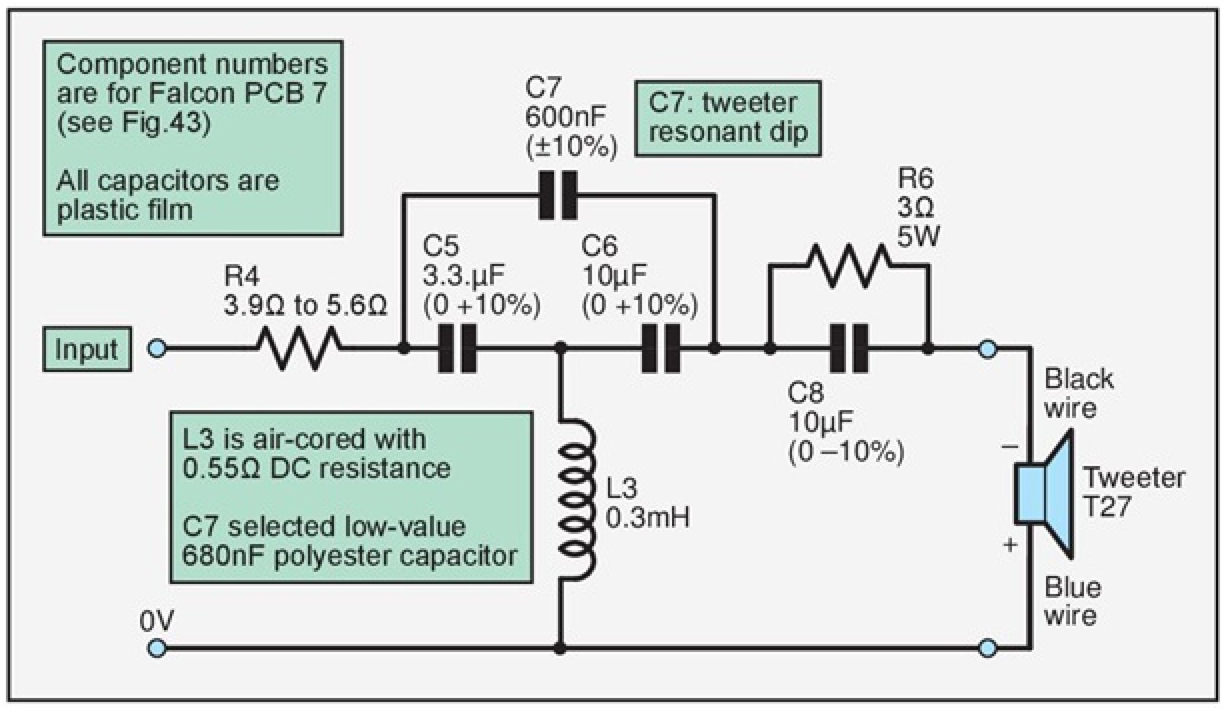

Component numbers
are for Falcon PCB 7
(see Fig.43)
All capacitors are
plastic film
C7
600nF
(±10%)
C7: tweeter
resonant dip
R6
3Ω
5W
R4
3.9Ω to 5.6Ω
C5
3.3.µF
(0 +10%)
C6
10µF
(0 +10%)
Input
C8
10µF
(0 –10%)
Black
wire
L3 is air-cored with
0.55Ω DC resistance
C7 selected low-value
680nF polyester capacitor
L3
0.3mH
Tweeter
T27
Blue
wire
0V

Fig.51. KEF T27 acoustic Butterworth filter, to be coupled with the 15Ω LS3/5A low-pass section shown in Fig.33. This was used in the KEF 104aB and the Falcon 23aB circuits.

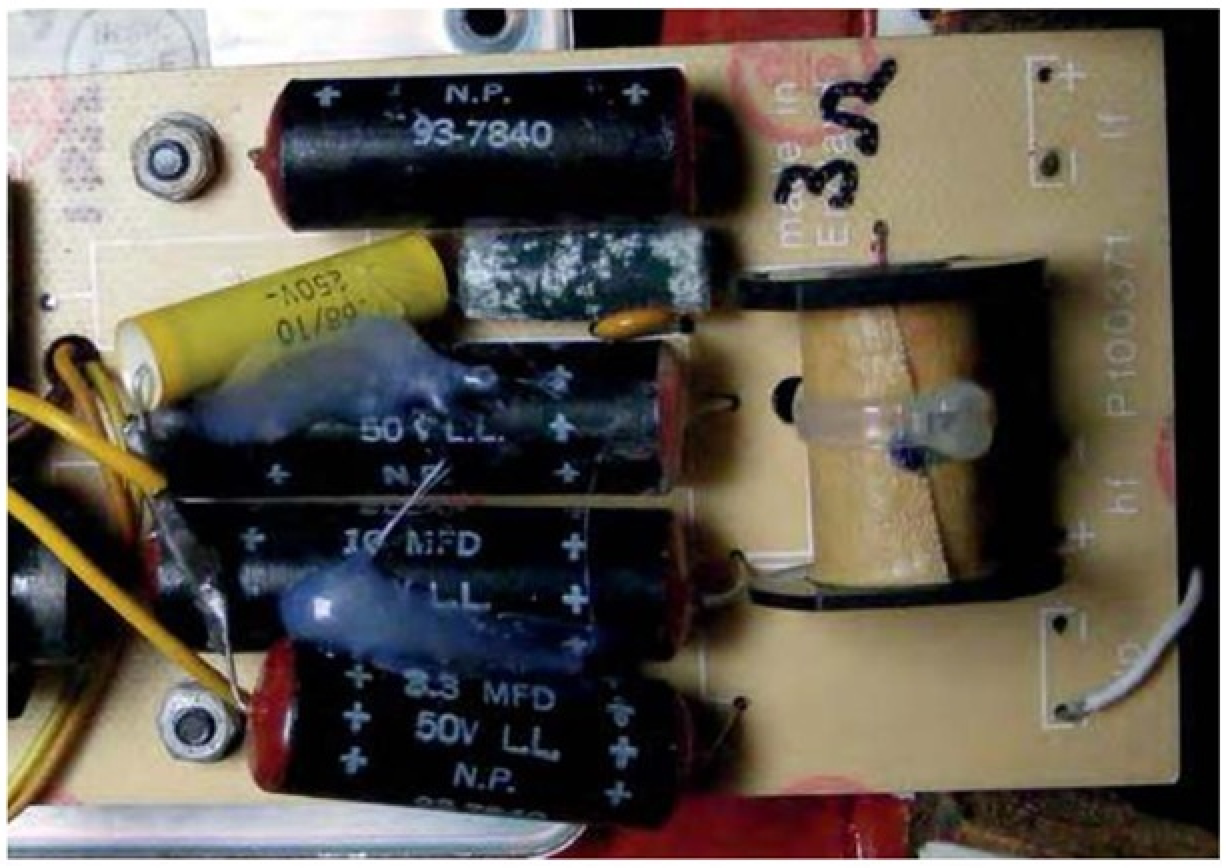

N.P.
93-7840
50 V L.L.
10 MFD
8.3 MFD
50V L.L.
N.P.
P100371

Fig.52. KEF 104aB crossover internals. Note the Callins/Elcap EX50 non-polarised capacitors. These are old wet electrolytics, but epoxy sealed so they last well. They should still always be checked for high ESR and raised capacitance values. In this case, an original 600nF device has been replaced with a yellow polyester type.

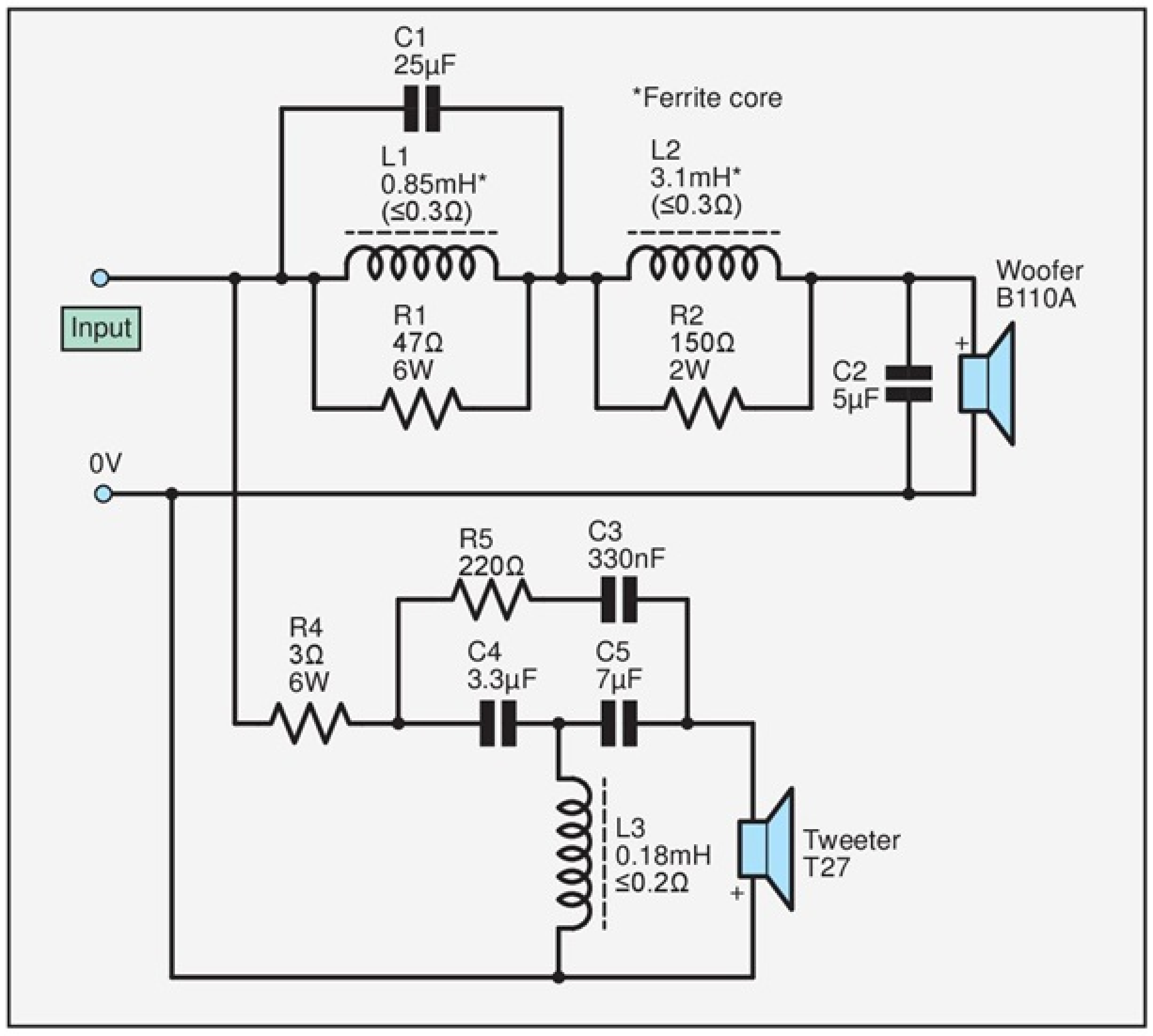


Fig.53. The KEF CS1A crossover. This has less of a mid-peak than the proper LS3/5A crossover. I prefer the CS1A.

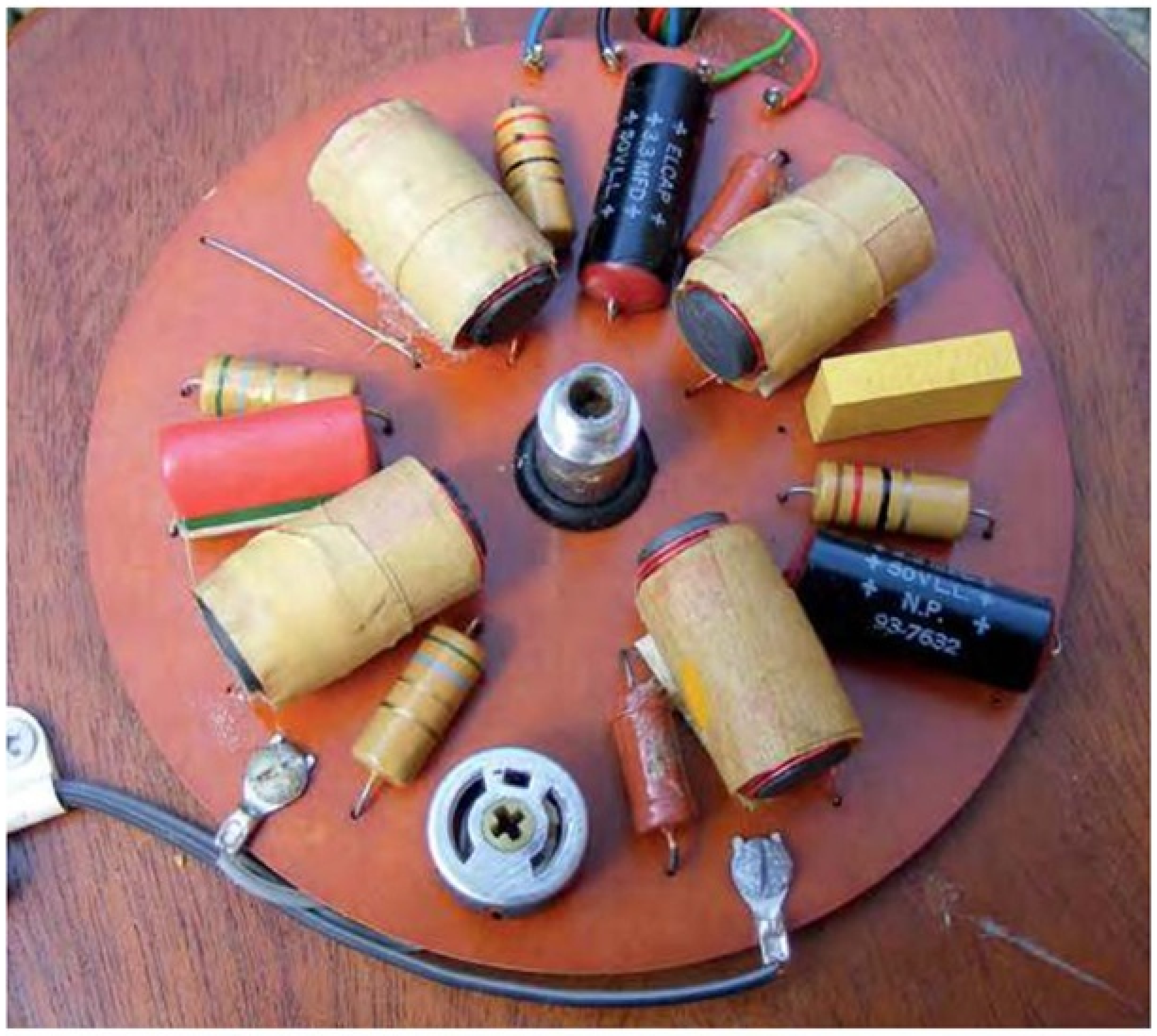


Fig.55. No, it's not a missile circuit board, but the circular crossover from a JR149.

effect becomes more intense as the frequency increases, so is worse in bass/mid drivers in two-way systems. It does not seem to be a problem with woofers below around 200Hz. Here, aluminium voice-coils do offer a worthwhile power handling advantage in large bass reflex systems because the damping effect limits 'jump-out' when the coil moves too far out of the magnetic gap.

Good news

One problem encountered by home constructors of Hi-Fi equipment is that old adage 'you can buy it ready-made at Richer Sounds for half the cost of buying the bits'. I have a classic solution for the beleaguered constructor who can't afford the Falcon monitor parts, and that is some nice surplus stock. This was kindly donated to me by the wife of Hi-Fi designer Charles Pepper, a retired speaker maker. I had a whole garage full of these excellent Beyond-The-Box speakers based on Danish Vifa units. I've kept 30 drivers and crossovers back for readers, although the bulk had to go to Earworm Records in York. They are happy to supply complete speakers, although you will have to paint the cabinets. More details, including simple tweaks that can be done to improve these speakers in later articles of this series.

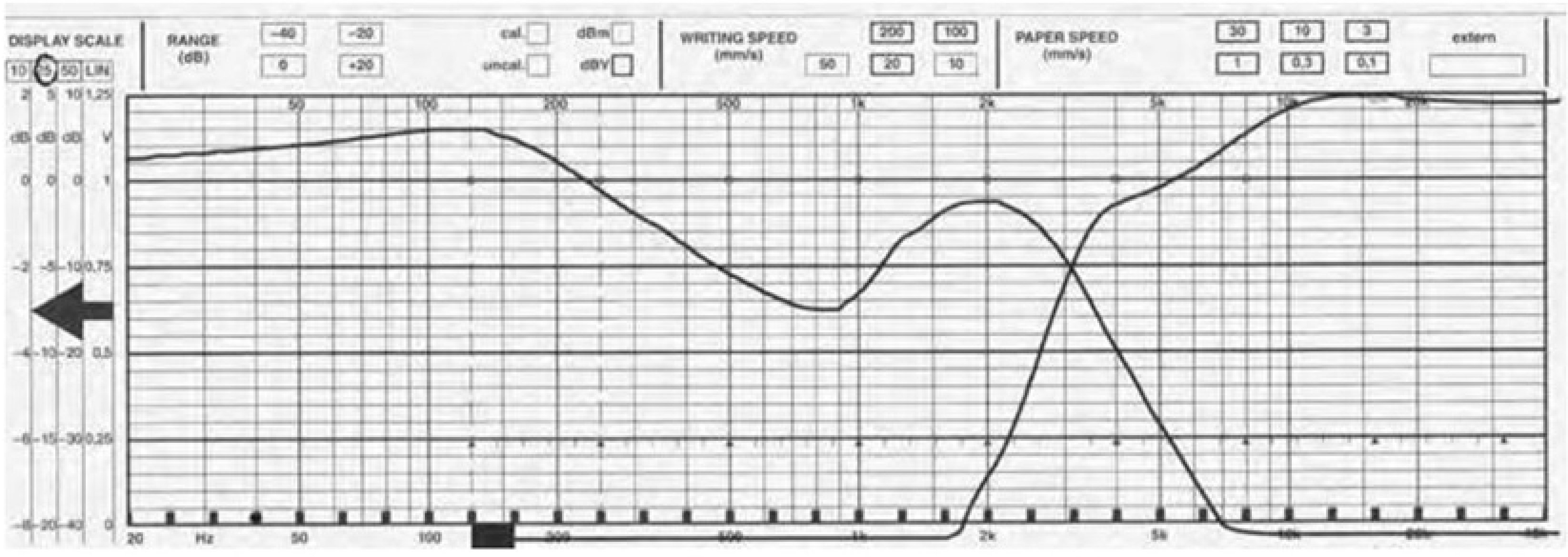


Fig.54. The electrical crossover curve of the BBC 15Ω LS3/5A using the drivers as loads.

Next month

We've covered a huge amount of LS3/5A-specific material in *Part 1* and *Part 2*. Next month, we take a short break to look at some general speaker hardware items (but still relevant to this project) before returning in the December issue with the design and construction of a passive LS3/5A design.

PE Mini-monitor resources

BBC R&D papers
www.bbc.co.uk/rd/publications

Drive units B110 data sheet and cut out details – see the October 2019 download on the *PE* website for data sheets for both the B110A and T27.
www.t-linespeakers.org/drivers/b110.html

Unofficial LS3/5A Support Site
This has a fantastic section 'library & links' with the BBC papers. The cabinet drawings by Peter Lotter can be found in the DIY section.
www.ls35a.com

Earworm records
www.earwormrecords.co.uk
earwormrecordsyork@gmail.com
01904 627488

Falcon Acoustics
www.falconacoustics.co.uk
01865 358001

Wavecor
www.wavecor.com

Good article: *A History of Kef Drive Units from the 1960s and 70s*
http://bit.ly/pe-oct19-kef

Wilmslow Audio
www.wilmslowaudio.co.uk

If you want to hear some electronic music mixed on LS3/5As.
https://soundcloud.com/shufflehound

Sealed box calculator
http://bit.ly/pe-oct19-boxcalc

Free speaker design software
http://bit.ly/pe-oct19-spkr

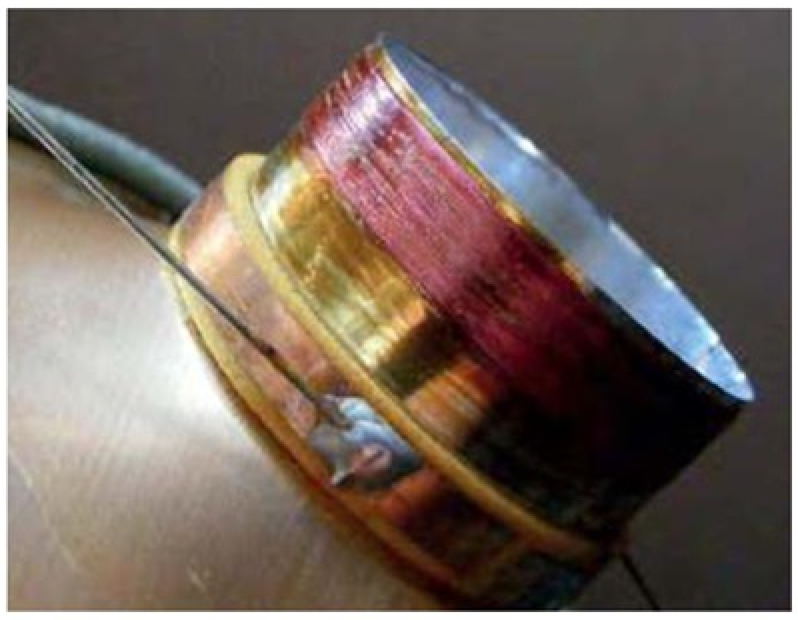


Fig.56. Voice-coil wound on an aluminium former. I suspect these cause problems with eddy-current induction.

ETI BUNDLE (1) Teach-In 3, 4 and 5 – all on CD-ROM – only £18.95

ELECTRONICS TEACH-IN 3 – CD-ROM

Mike & Richard Tooley

The three sections of the *Teach-In 3* CD-ROM cover a huge range of subjects that will interest everyone involved in electronics – from newcomers to the hobby and students to experienced constructors and professionals.

The first section (80 pages) is dedicated to *Circuit Surgery*, *EPE/PE's* regular clinic dealing with readers' queries on circuit design problems – from voltage regulation to using SPICE circuit simulation software.

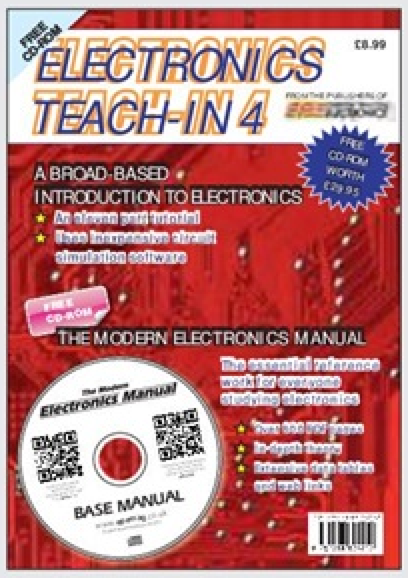
The second section – *Practically Speaking* – covers hands-on aspects of electronics construction. Again, a whole range of subjects, from soldering to avoiding problems with static electricity and identifying components is covered. Finally, our collection of *Ingenuity Unlimited* circuits provides over 40 circuit designs submitted by readers.

The CD-ROM also contains the complete *Electronics Teach-In 1* book, which provides a broad-based introduction to electronics in PDF form, plus interactive quizzes to test your knowledge and TINA circuit simulation software (a limited version – plus a specially written TINA Tutorial).

The *Teach-In 1* series covers everything from electric current through to microprocessors and microcontrollers, and each part includes demonstration circuits to build on breadboards or to simulate on your PC.

ELECTRONICS TEACH-IN 4 – CD-ROM
A BROAD-BASED INTRODUCTION TO ELECTRONICS

Mike & Richard Tooley



The *Teach-In 4* CD-ROM covers three of the most important electronics units that are currently studied in many schools and colleges. These include, Edexcel BTEC level 2 awards and the electronics units of the Diploma in Engineering, Level 2.

The CD-ROM also contains the full *Modern Electronics Manual*, worth £29.95. **The Manual** contains over 800 pages of electronics theory, projects, data, assembly instructions and web links.

A package of exceptional value that will appeal to anyone interested in learning about electronics – hobbyists, students or professionals.

ELECTRONICS TEACH-IN 5 – CD-ROM
JUMP START

Mike & Richard Tooley

15 design and build circuit projects for newcomers or those following courses in school and colleges.

The projects are: ■ Moisture Detector ■ Quiz Machine ■ Battery Voltage Checker ■ Solar-Powered Charger ■ Versatile Theft Alarm ■ Spooky Circuits ■ Frost Alarm ■ Mini Christmas Lights ■ iPod Speaker ■ Logic Probe ■ DC Motor Controller ■ Egg Timer ■ Signal Injector Probe ■ Simple Radio Receiver ■ Temperature Alarm.

PLUS

PIC'n' Mix – starting out with the popular range of PIC microcontrollers and *Practically Speaking* – tips and techniques for project construction.

The CD-ROM also contains:

- ■ Complete *Teach-In 2* book, a practical introduction to PIC microprocessors
- ■ MikroElektronika, Microchip and L-Tek PoScope software.

Three Teach-ins for the great price of £18.95

PLUS you also get the contents of the free CD-ROM from each issue...

... so that's another TWO Teach-Ins and The Full Modern Electronics Manual!

What a Bargain!!

ETI BUNDLE (2) Teach-In 6, 7 and 8 – all on CD-ROM – only £18.95

ELECTRONICS TEACH-IN 6 – CD-ROM
A COMPREHENSIVE GUIDE TO RASPBERRY Pi

Mike & Richard Tooley

Teach-In 6 contains an exciting series of articles that provides a complete introduction to the Raspberry Pi, the low cost computer that has taken the education and computing world by storm.

This latest book in our *Teach-In* series will appeal to electronics enthusiasts and computer buffs who want to get to grips with the Raspberry Pi.

Teach-In 6 is for anyone searching for ideas to use their Pi, or who has an idea for a project but doesn't know how to turn it into reality. This book will prove invaluable for anyone fascinated by the revolutionary Pi. It covers:

- ■ Pi programming
- ■ Pi hardware
- ■ Pi communications
- ■ Pi Projects
- ■ Pi Class
- ■ Python Quickstart
- ■ Pi World
- ■ ...and much more!

The *Teach-In 6* CD-ROM also contains all the necessary software for the series, so that readers and circuit designers can get started quickly and easily with the projects and ideas covered.

ELECTRONICS TEACH-IN 7 – CD-ROM
DISCRETE LINEAR CIRCUIT DESIGN

Mike & Richard Tooley

Teach-In 7 is a complete introduction to the design of analogue electronic circuits. It is ideal for everyone interested in electronics as a hobby and for those studying technology at schools and colleges. The CD-ROM also contains all the circuit software for the course, plus demo CAD software for use with the *Teach-In* series.

- ■ Discrete Linear Circuit Design
- ■ Understand linear circuit design
- ■ Learn with 'TINA' – modern CAD software
- ■ Design simple, but elegant circuits
- ■ Five projects to build:
  - i) Pre-amp
  - ii) Headphone Amp
  - iii) Tone Control
  - iv) VU-meter
  - v) High Performance Audio Power Amp.

PLUS

Audio Out – an analogue expert's take on specialist circuits

Practically Speaking – the techniques of project building.

ELECTRONICS TEACH-IN 8 – CD-ROM
INTRODUCING THE ARDUINO

Mike & Richard Tooley

Hardware: learn about components and circuits
Programming: powerful integrated development system
Microcontrollers: understand control operations
Communications: connect to PCs and other Arduinos.

Teach-In 8 is an exciting series designed for electronics enthusiasts who want to get to grips with the inexpensive, popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. It will provide a one-stop source of ideas and practical information.

The Arduino offers a truly effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle through wireless or the Internet.

Teach-In 8 is based around a series of practical projects with plenty of information for customisation.

This book also includes *PIC n' Mix*: 'PICs and the PICkit 3 – A Beginners guide' by Mike O'Keefe and *Circuit Surgery* by Ian Bell – 'State Machines part 1 and 2'.

The CD-ROM includes the files for:

- ■ *Teach-In 8*
- ■ Microchip MPLAB IDE XC8 8-bit compiler
- ■ PICkit 3 User Guide
- ■ Lab-Nation Smartscope software.

ORDER YOUR BUNDLE TODAY!
JUST CALL 01202 880299 – OR VISIT www.epemag.com

FD&19MXZ

STORE YOUR BACK ISSUES ON CD-ROMS

VOL 1 **BACK ISSUES** – January 1999 to June 1999
Plus bonus material from Nov and Dec 1998
VOL 2 **BACK ISSUES** – July 1999 to December 1999
VOL 3 **BACK ISSUES** – January 2000 to June 2000
VOL 4 **BACK ISSUES** – July 2000 to December 2000
VOL 5 **BACK ISSUES** – January 2001 to June 2001
VOL 6 **BACK ISSUES** – July 2001 to December 2001
VOL 7 **BACK ISSUES** – January 2002 to June 2002
VOL 8 **BACK ISSUES** – July 2002 to December 2002
VOL 9 **BACK ISSUES** – January 2003 to June 2003
VOL 10 **BACK ISSUES** – July 2003 to December 2003
VOL 11 **BACK ISSUES** – January 2004 to June 2004
VOL 12 **BACK ISSUES** – July 2004 to December 2004
VOL 13 **BACK ISSUES** – January 2005 to June 2005
VOL 14 **BACK ISSUES** – July 2005 to December 2005
VOL 15 **BACK ISSUES** – January 2006 to June 2006
VOL 16 **BACK ISSUES** – July 2006 to December 2006
VOL 17 **BACK ISSUES** – January 2007 to June 2007
VOL 18 **BACK ISSUES** – July 2007 to December 2007
VOL 19 **BACK ISSUES** – January 2008 to June 2008
VOL 20 **BACK ISSUES** – July 2008 to December 2008
VOL 21 **BACK ISSUES** – January 2009 to June 2009
VOL 22 **BACK ISSUES** – July 2009 to December 2009

ONLY £16.45 each
Five years just £29.95
including VAT and p&p

VOL 23 **BACK ISSUES** – January 2010 to June 2010
VOL 24 **BACK ISSUES** – July 2010 to December 2010
VOL 25 **BACK ISSUES** – January 2011 to June 2011
VOL 26 **BACK ISSUES** – July 2011 to December 2011
VOL 27 **BACK ISSUES** – January 2012 to June 2012
VOL 28 **BACK ISSUES** – July 2012 to December 2012
VOL 29 **BACK ISSUES** – January 2013 to June 2013
VOL 30 **BACK ISSUES** – July 2013 to December 2013
VOL 31 **BACK ISSUES** – January 2014 to June 2014
VOL 32 **BACK ISSUES** – July 2014 to December 2014
VOL 33 **BACK ISSUES** – January 2015 to June 2015
VOL 34 **BACK ISSUES** – July 2015 to December 2015
VOL 35 **BACK ISSUES** – January 2016 to June 2016
VOL 36 **BACK ISSUES** – July 2016 to December 2016
VOL 37 **BACK ISSUES** – January 2017 to June 2017
VOL 38 **BACK ISSUES** – July 2017 to December 2017
VOL 39 **BACK ISSUES** – January 2018 to June 2018
VOL 40 **BACK ISSUES** – July 2018 to December 2018

FIVE-YEAR DVD/CD-ROMs

No.1 – Jan 03 to Dec 07
No.2 – Jan 04 to Dec 08
No.3 – Jan 05 to Dec 09
No.4 – Jan 06 to Dec 10
No.5 – Jan 07 to Dec 11
No.6 – Jan 08 to Dec 12
No.7 – Jan 09 to Dec 13
No.8 – Jan 10 to Dec 14
No.9 – Jan 11 to Dec 15
No.10 – Jan 12 to Dec 16
No.11 – Jan 13 to Dec 17
No.12 – Jan 14 to Dec 18 – **NEW!**

15 YEARS OF EPE!

AMAZING
BUNDLE OFFER

No.1 – Jan 03 to Dec 07
PLUS
No.6 – Jan 08 to Dec 12
PLUS
No.11 – Jan 13 to Dec 17

ALL FOR JUST
£39.95

All DVD/CD-ROMs are suitable for any PC with a DVD/CD-ROM drive.
They require Adobe Acrobat Reader (free from: www.adobe.com/acrobat)

ORDER YOUR CD-ROMs TODAY!
JUST CALL 01202 880299 – OR VISIT www.epemag.com

Make it with Micromite

Phil Boyce – hands on with the mighty PIC-powered, BASIC microcontroller

Part 9: Introduction to MMBASIC graphical commands

ED&19MX77

Last month, in Part 8, we discussed how to assemble the versatile *IPS Display Module* (*IDM*). We also covered how to check that it had been assembled correctly by loading a test program into the *MKC*; this resulted in a simplified clock being displayed on the IPS screen, as shown in Fig.1.

The *IDM* comprises four useful elements:

- Mini IPS colour display
- Accurate real-time clock (RTC) module
- Infrared (IR) receiver
- Piezo sounder.

With these four elements connected to a Micromite, the *IDM* forms the perfect basis for our *Mini MKC Clock* project; and by only changing the program in the Micromite, the *IDM* can also be used as the hardware base for many other different projects.

In previous parts of this series we have discussed how to interact with an IR receiver, and also a piezo sounder. Therefore, to be able to write programs that fully utilise all of the *IDM's* hardware we need to learn how to use MMBASIC's graphical commands; and this will be the main topic covered this month. We will learn by example, demonstrating how to use the commands that write and draw to the mini IPS colour display. We will also discuss the two commands that interact with the RTC module. To assist with our learning, we will first go over the details of the *IDM* test program from last month, and how we can 'tidy up' the code by using what is called the 'Library' to store the IPS screen driver. Finally, to wrap everything up, we will load a new version of the *Mini MKC Clock* program into the *MKC*, and discuss some additional ideas and features you may wish to implement into your own customised version of the code.

Questions? Please email Phil at:
contactus@micromite.org

The *IDM* introduced last month will be used throughout the topics covered here (and also for future projects). So, if you haven't already done so, we recommended you first take the time to build an *IDM* – this will also help you appreciate just how easy it is to create a project which incorporates a high-quality IPS screen.

Preparation

In order to set things up for this month, you will need to successfully install and run the *IDM* test program (as outlined at the end of *Part 8*).

To recap the steps involved, first plug the *IDM* into the *DM*/*MKC*, and connect this to your computer. Start your terminal application (TeraTerm or PuTTY for Windows; Terminal/Screen for MacOS) and ensure that you have the **IDM_TestCode.txt** file installed in your *MKC* (download from the September page of the *PE* website) – note that you may still have this in your *MKC* if you followed last month's article. Press the reset button on the *DM*, check to see if the IPS display shows a simplified clock. If not, type `RUN` at the command prompt to start the program (ignore any 'CPU exception' errors), and then press the reset button once again. You should now see the clock; if not, simply work through the detail of the test procedure discussed at the end of last month's article.

Important Note – some people have reported that the *IDM* test program displays the clock but that it is drawn in the wrong colours (and is offset slightly) as shown in Fig.2. If this is the case, then you will need to change the last three parameters in the line of code within `SUB mm.startup` (which is at the end of the program, and which is also highlighted in bold in Listing 1). Make the change as follows:

```
(from): ST7735S 22,23,21, 1,1,-1,-2
(to): ST7735S 22,23,21, 1,0,0,0
```

Once the *IDM* test program is successfully running and the colours are correct, press `Ctrl-C` to stop the test program. Notice that the clock is still displayed on the IPS screen – but, a closer look reveals the time is not being updated.

The IDM test program

The *IDM* test code is currently a single program; this was intentional to help make last month's testing easier. However, this test program should really be considered as comprising two main 'blocks' of code. Referring to Listing 1, and to help with the following discussion, there are two blocks of code: one without a shaded background and one highlighted with a shaded background to indicate the following:

- The unshaded block of code (ie, the top part of the listing) is the actual program code. This code redraws the time and date (continually) on the IPS screen, synchronises the Micromite with the RTC (every minute), and 'beeps' the piezo sounder (whenever an IR signal is detected).
- The shaded code is the IPS screen driver. This is an essential (and pre-written) piece of code and ensures the IPS screen works properly and responds correctly to MMBASIC's graphical commands. You do not need to worry about how this code works – just accept that it does all the 'difficult stuff' required to make the attached IPS screen work.

If you now start the EDITor (`F4`) you will see the **IDM_TestCode** program; it should be identical to that shown in Listing 1 (note that Listing 1 has a chunk of the shaded IPS screen driver code missing – this is just to keep the printed listing reasonably short). Starting at the top, and working down towards the line: `'IPS DRIVER,`

Micromite code

The code in this article is available for download from the *PE* website.

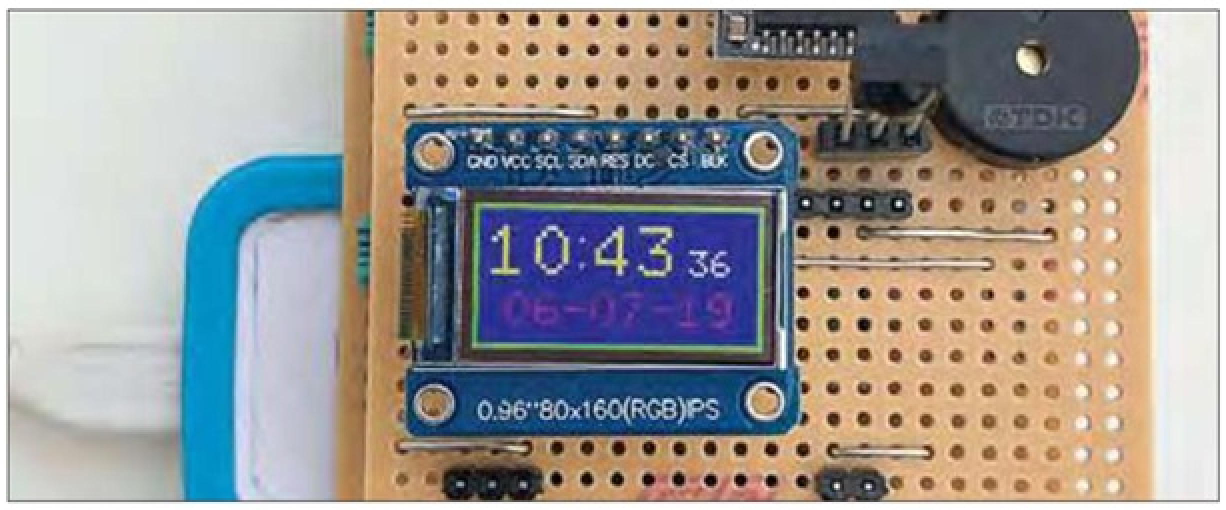


Fig.1. Running the IDM test program will display this simple clock on the IPS screen.

you should recognise all of the commands used apart from three (don't worry if you don't understand what the program is doing – it will be explained shortly). The three commands not covered so far are: `BOX`, `TEXT` and `RTC GETTIME`. These will be explained later in this article; they are commands that control graphics and the RTC.

Scroll through the rest of the *IDM* test code (below the line: `'IPS DRIVER`) and you will see lines of alphanumeric characters. This is the screen driver and will be unintelligible, but do not alter it in any way! (It is a hexadecimal representation of some C code – but this is a topic that is beyond the scope of this series.)

The screen driver

The important point to understand is that the screen driver is an essential block of pre-written code that in simple terms makes it possible to use MMBASIC's graphical commands with an attached screen. We are not talking here about the console screen – we are referring to a dedicated graphical screen to which the Micromite can write text and numbers, and draw graphics. MMBASIC's graphical commands do not communicate with the console screen (but `PRINT` does, as do VT100 Escape codes discussed last month).

There are many different types of graphical screens that we can connect with a Micromite; however, only one type of graphical screen can be connected at a time. The choice really depends on which screen best suits the requirements of a project. Different screens have different properties, the most obvious are its physical size and the screen's pixel resolution. On a technical level, each screen will have a different, yet essential, 'start-up configuration sequence'. This is a sequence of data values that need to be sent to the screen's dedicated built-in controller chip in order to define certain properties (including screen size, pixel count, orientation, and type of hardware interface). The start-up sequence is an important part of the screen driver code and is sent from the Micromite to the attached display whenever power is first applied (get the sequence of the startup-data wrong, and the screen simply won't work). So the next important point to understand is that each type of screen needs its own specific pre-written screen driver to work properly; there is no one generic screen driver that will work with every type of graphical screen.

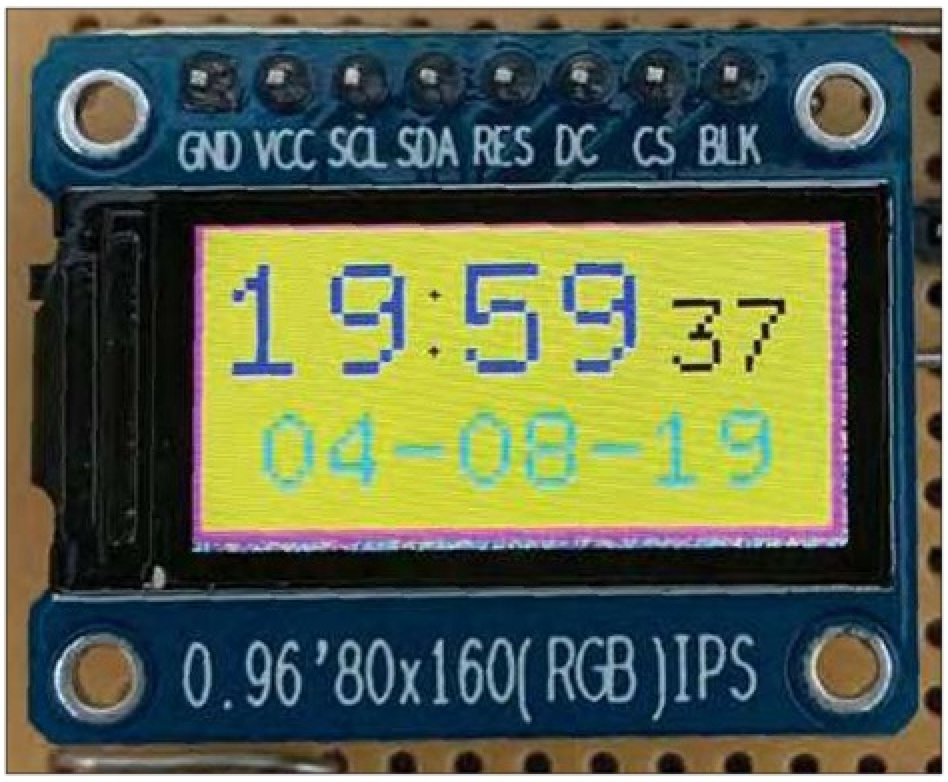


Fig.2. If your clock is displayed with incorrect colours (and is offset) as shown here, then change the last three parameters in `SUB MM.STARTUP` (see 'Important Note' opposite).

That said, there is a family of screens that all use the ILI9341 IC. The ILI9341 typically controls LCD screens ranging in size from 2.2 up to 3.2-inches, and all with a pixel resolution of 320 x 240. These ILI9341 screens can actually use the same ILI9341 screen driver; so for this reason, this 'common' screen driver is built into the MMBASIC firmware. To enable it, you use the command `OPTION LCDPANEL ILI9341,...` However, there are many other different types of graphical screen that you may wish to use instead, and to have every single screen driver built into MMBASIC would consume much too much memory. For this reason, the specific screen driver that you require (when not using an ILI9341 screen) will first need to be installed (ideally in the method that we will now discuss).

The screen driver should be regarded as a block of pre-written code that never needs to be changed. As it currently stands in the *IDM* test program, the IPS screen driver is at risk of being altered (accidentally or otherwise) from within the EDITor; for example, a slip of the hand on the keyboard could result in just one of the many lines of hexadecimal numbers being changed; and even if it is only changed by just a single character, the result will almost certainly make the IPS screen unresponsive (and possible the Micromite too). To avoid this we are now going to learn how to keep the screen driver code out of harm's way by placing it in a part of the Micromite's memory called the 'Library'.

The Library – a safe place to be

The Library is a special area in the Micromite that can be thought of as a safe place to store code that does not need to be altered in any way. This code is permanently stored in the Micromite's memory – even after typing `NEW` to delete a program. Understand that code stored in the Library will not appear in the EDITor when editing a program (which is exactly why the Library is a safe place for the code to be), but it is still accessible to the program code. Some examples of code that are suited for the Library include:

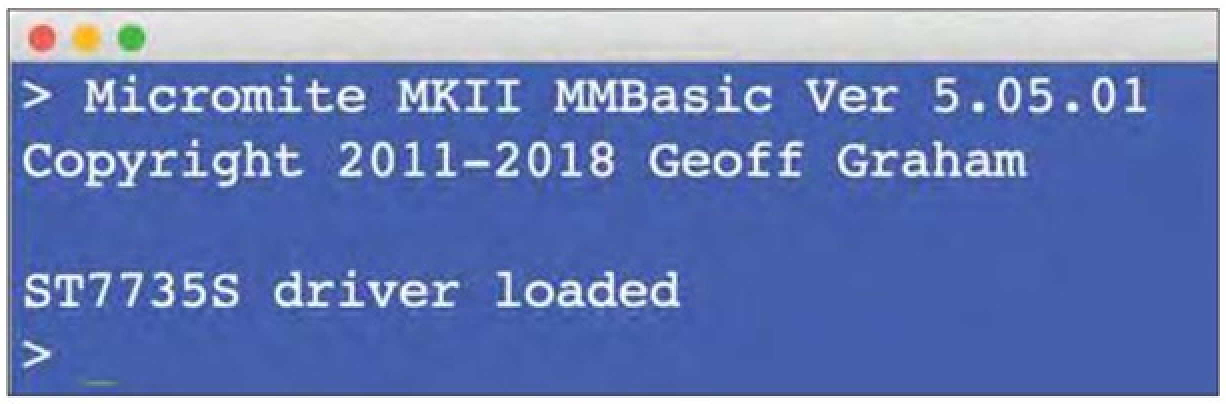
- Drivers (as currently being discussed)
- User-defined fonts (a future topic)
- SUBs you've written which add new commands to MMBASIC

Note that a subroutine can be written in the C language (instead of the more usual BASIC); and in this case it is called a `CSUB`. Any subroutine written in C will run much faster than one written in BASIC; and a `CSUB` is visible in MMBASIC code as a sequence of hexadecimal numbers (generated by an external program). If you look at the IPS screen driver in the *IDM* test program you will see that it is a `CSUB`.

To add code to the Library, you first enter it into the EDITor as you would do with a normal program, and then RUN it to test that it works as expected. Once you are happy it is working correctly, you simply type `LIBRARY SAVE` at the command prompt. This will then compress the code, and move it into the Library. The program memory will then be emptied (since the code is now safe within the Library). Note that `LIBRARY SAVE` can be used multiple times – each time, any code in the program memory (EDITor) will be added to the end of any existing code in the Library. We will now put the IPS screen driver into the Library – take care to follow the steps exactly as discussed below.

First, we need to remove the parts of the code from the **IDM_TextCode** program that are not part of the screen driver (in other words, delete the lines of code that are unshaded in Listing 1). Simply go to the EDITor and delete the top section of the program to remove the initial comments, the Setup, the Main-Program, and the three SUBs – this will leave everything from the line `' IPS DRIVER` down to the end (do not shorten the `CSUB` section in any way).

Now press F2 to SAVE and RUN the code. Check you get no syntax errors. Next, press the reset button and check that after the usual MMBASIC 'welcome message' you see the line `ST7735S driver loaded` (as shown in Fig.3).



```
> Micromite MKII MMBasic Ver 5.05.01
Copyright 2011-2018 Geoff Graham

ST7735S driver loaded
> _
```

Fig.3. On power-up, you should see the above confirmation message indicating that the ST7735S screen driver has been successfully loaded.

If you get any syntax error, or you do not see the line `ST7735S driver loaded`, then repeat the above steps of loading **IDM_TestCode.txt**, and the subsequent editing to extract the IPS screen driver, taking care not to alter the screen driver code in any way.

Once this is operating correctly, type `LIBRARY SAVE` (at the command prompt). On pressing Enter, the Micromite will store a compressed version of the code into the Library. To check this, two things are worth mentioning that you should try out. First, go into the EDITor – it should now be blank with no code visible (if not – then check for any missed error messages in the steps above). Exit the EDITor. The second check is to view the code in the Library with the command `LIBRARY LIST`. This will show the compressed version of the IPS screen driver – see Fig.4 (note that the hexadecimal numbers between the lines `CSub ST7735S` and `End CSub` have been compressed and hence are no longer visible).

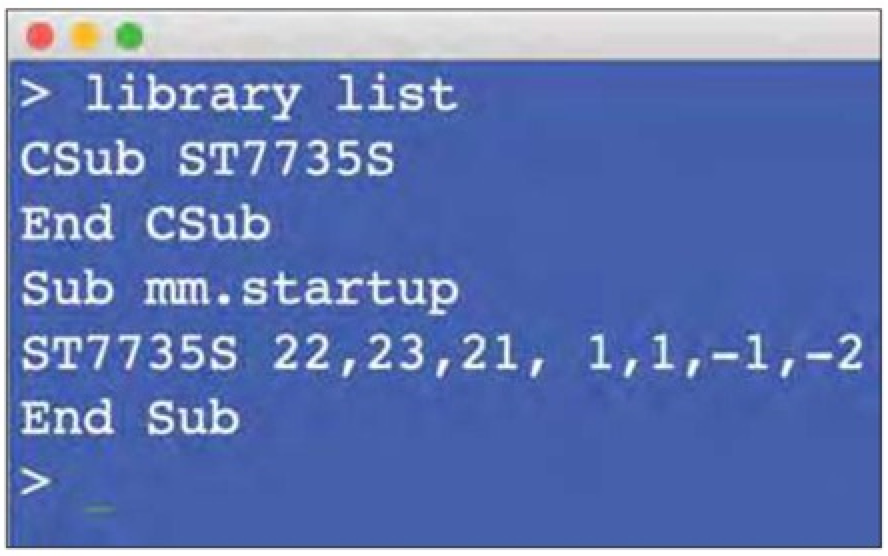
You have now successfully loaded the IPS screen driver into the Micromite's Library where it will remain in place until you purposely delete it. Just for your information, to remove the screen driver, you simply delete all the code stored in the Library. This is achieved with the command `LIBRARY DELETE` (don't do this now). One more useful tip: if you need to check if there is code in the Library, simply type `LIBRARY LIST` at the command prompt. This is handy because sometimes any existing (and unrequired) Library code that you have forgotten about may prevent you from accessing an I/O pin due to the Library code reserving certain I/O pins. To demonstrate this, simply type the following at the command prompt: `SETPIN 22,DOUT` – you will see an error message warning you that pin 22 is reserved on startup (it is used by the IPS screen driver).

One more very important point that needs to be discussed here is the subroutine `SUB mm.startup` in the last three lines of code.

SUB MM.STARTUP

This is really simple to follow – I promise! If a `SUB` is given the reserved name `MM.STARTUP` then the code contained within that `SUB` will be run on initial power-up (before the program in the main memory is even run). One very good example as to why you may need to do this is to load a screen driver – exactly as here. Note that `SUB MM.STARTUP` can be placed either in the Library, or in normal program memory; however, in most cases it makes sense to test it first in program memory, and then once it is working as expected, place `SUB MM.STARTUP` in the Library so that nothing is accidentally altered.

Assuming you have successfully followed the previous steps, then `SUB MM.STARTUP` will be in the Library. On powering up the *MKC*, the single line of code we have here in `SUB MM.STARTUP` is run, and this is just a call to the subroutine named `ST7735S` (which is a `CSUB` containing the pre-written IPS screen driver). The numbers you can see alongside are (configurable) parameters that `CSUB ST7735S` needs to be informed about. For those interested in a little more detail: the first three numbers are the I/O pin numbers that connect the Micromite with the IPS display (to IPS module pins DC, RES, CS). Next is a value between 1 and 4 that represents the screen's orientation (ie, landscape or portrait, normal or inverted). The last three numbers are 'tweaks' to alter the image on the screen to allow for variations across different IPS module manufacturers (see 'Important Note' in the 'Preparation' section above). The first of these three numbers is either a 0 or a 1, and will result in the binary representation of the colour values being 'logically inverted' (put another way, if the colours are wrong, change this value). The last two parameters are an offset so that the top-left corner of the screen can be set as coordinate 0,0 (more on coordinates later).



```
> library list
CSub ST7735S
End CSub
Sub mm.startup
ST7735S 22,23,21, 1,1,-1,-2
End Sub
> _
```

Fig.4. To view any code in the Library, use the command `Library List`. Here we see the compressed version of the IPS screen driver code.

And that's all there is to it – we have loaded the required pre-written IPS screen driver 'C' code into the (safe) Library where it can't be accidentally altered; and used a `SUB` named `MM.STARTUP` to ensure that the screen driver is loaded on power up.

'Tidier' version of IDM_TestCode

Since we now have the IPS screen driver safely stored in the Library, we can remove its code from the original *IDM* test program (ie, remove the shaded code in Listing 1). This produces a much shorter (and hence tidier) version of the **IDM_TestCode** program.

The *MKC* should now have no code in the program memory (wiped clear after performing the `Library Save`). Reload the original **IDM_TestCode.txt** file into the *MKC*, and then use the EDITor to remove every line of code after the last SUB in the Subroutines section (hence delete all the lines from `' IPS DRIVER` to the very end). Press `F2` to SAVE and RUN, and the code should run as it did before – if not, check the code in the EDITor is the same as that shown unshaded in Listing 1.

Let's just take a moment to understand that all we have done is to split the initial *IDM* test program into two parts – the screen driver 'chunk' is stored safely in the Library (this code doesn't need to change), and the program code 'chunk' is stored in the program memory (we can still edit the code's functionality in the same way we would alter any program).

How the IDM test code works

With this shortened version of the program, it is much easier to explain how it works. For the following explanation, you can either refer to the unshaded code in Listing 1, or look at your code in the EDITor. As previously mentioned, all commands (apart from `BOX`, `TEXT`, and `RTC GETTIME`) have been discussed in earlier parts of this series.

We will now explain the functional lines of code (do also refer to the comments included at the end of each line of code), and remember that this code is simply showing the time (and date) on the IPS screen.

The first line of code in the SETUP section of the program sets `OPTION AUTORUN ON`, ensuring that the program will automatically run on power up (no need to manually type `RUN` at the command prompt). Remember, this only takes effect after the first time the program is run (unless you have already set `OPTION AUTORUN ON` at the command prompt).

The next line sets up an IR interrupt subroutine (named `IR_Int`) which will automatically be called whenever the Micromite detects an IR signal on input Pin 16 (the pin to which the IR receiver outputs any IR data detected).

A subroutine (`SUB TimeFromRTC`) is set to be called automatically every minute (60000ms = 60s = 1min) with the `SETTICK` command.

The fourth line of code immediately calls the subroutine `TimeFromRTC`, which in turn loads the Micromite's `Time$` and `Date$` with the high-accuracy data stored in the RTC module (by use of the `RTC GETTIME` command that will be explained shortly).

The `PWM` line of code performs two simultaneous tasks (on PWM Channel 2, at a frequency of 800Hz). PWM Channel 2A (Micromite pin 26) is set to 100% duty cycle and this turns on the IPS backlight to maximum brightness. PWM Channel 2B (Micromite pin 24) is simultaneously set to 50% duty cycle resulting in an 800Hz square-wave signal on one side of the piezo sounder. Note that the piezo sounder will not make a sound until its other pin (connected to Micromite pin 15) is at 0V. The next line of code is using the `BOX` command, and this functionality will be explained in detail shortly (in brief, it clears the screen by drawing a blue box with a green border).

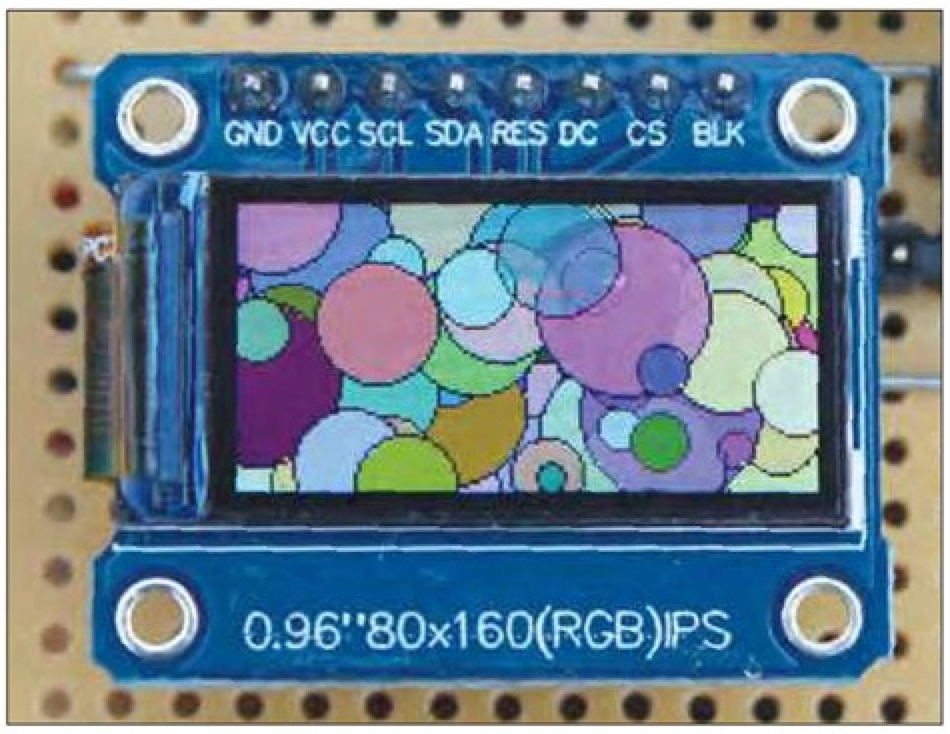


Fig.5. `GUI TEST LCDPANEL` produces this animation drawing lots of circles. It is useful to check that the attached screen is working, and also that the screen driver is correctly installed.

Next, a subroutine (`colon`) is called; this in turn uses two `TEXT` commands (that will also be explained shortly); in essence, it draws two separate dots which form the ':' character between the clock's hours and minutes.

Next comes the `MAIN PROGRAM` section, which is just a `DO/LOOP` containing four lines. Each of these four lines of code uses the `TEXT` command to continually display the current time and date.

The only other code left to explain is the three lines of code in `SUB IR_Int` which is called whenever an IR signal is detected. This subroutine (when called) initially sets Micromite pin 15 to an output, which by default is set low (ie, set to 0V). This places 0V onto the second pin of the piezo sounder, enabling it to make 'noise' (at 800Hz).

The next line of code is just a `PAUSE` for 10ms, after which the final line of code will set Micromite pin 15 back to a 'floating' state – the piezo sounder will be 'silenced'. The end result of this subroutine is a simple 'beep' whenever an IR signal is detected – and here in the test code it is used to confirm that both the piezo sounder and IR receiver are working properly (and that those parts of the *IDM* have been assembled correctly).

If the above still seems a little tricky to understand, just think of the program as continually updating the IPS display with the time and date from the Micromite's built-in `Time$` and `Date$`. Every 60 seconds, the `Time$` and `Date$` in the Micromite (which can drift very slightly) are resynchronised with the high-accuracy RTC. Whenever an IR signal is detected, a short beep sound is heard.

With the IPS screen driver safely installed in the Library, along with a better understanding of how the *IDM* test program operates, we can now begin exploring MMBASIC's graphical commands. We can actually do this by typing the command(s) directly at the command prompt (no need to remove, edit, or replace the *IDM* test program). But the really nice thing about all this is that with the *IDM's* high-quality IPS screen, we will immediately see the result of each command.

So before we start, ensure you have completed all the steps above, and then press Ctrl-C to stop the *IDM* test program from running – this should mean that you can see the command prompt in your terminal program.

MMBASIC graphical commands

Let's start by using a built-in graphics command to test that a connected screen does indeed work correctly. At the command prompt type `GUI TEST LCDPANEL` (and press Enter). You will now see the IPS screen fill up with rapidly drawn coloured circles – see Fig.5. Note that the command prompt on the console is unavailable while this test animation is playing on the IPS screen. To stop the test (and to get the command prompt back) simply press any key on the keyboard. Pressing a key will also clear the IPS screen by turning it back to black.

Now repeat the above command, but this time press `Ctrl-C` to stop the animation. You will see now that the circles remain on the screen (rather than being cleared). We will use these circles for the next topic.

IPS backlight control

When designing a project that incorporates a display, sometimes the display may need to be switched off (to save power, or because no screen information needs to be displayed). Likewise, at some other time, the display will need to be turned on. Rather than just abruptly switching the screen between dark (off) and bright (on), a nice effect is a smart-looking fade-up/fade-down. To achieve, this we need to be able to adjust the backlight's brightness; and this is why the *IDM* has been designed with the screens backlight (BLK pin) connected to a PWM output on the Micromite (PWM channel 2A; pin 26).

By default, the backlight will be fully on when the BLK pin on the IPS screen is either floating or is at 3.3V. However, by sending (low-going) pulses to PWM channel 2A, we can adjust the brightness of the backlight.

With the 'static' coloured circles that are currently on the screen, enter the following PWM command (being sure to look at the IPS display when you hit the Enter key): `PWM 2,800,50,50` – this sends a 50% duty cycle (ie, a square-wave) to the backlight; so it is rapidly (800Hz) switched between on/off. The result is that you should see the IPS backlight brightness reduce slightly. If you missed it, then repeat the above with a lower value in place of the highlighted `50` (for example, use: `PWM 2,800,20,50`). Repeat with `10`, and then `0`. You will see that the backlight becomes darker and darker, until it eventually turns off (with `0`). The reason it turns off with a PWM value of `0` is that the PWM output is effectively always at 0V (and taking the BLK pin low (0V) means the backlight is turned off). So by altering the relevant parameter of the PWM command (for channel 2A) we can control the IPS backlight brightness. Now set it back to full brightness with `PWM 2,800,100,50` and the circles will re-appear. One thing to point out here is that the value used (think of it as a percentage value between 0 and 100) is not a linear representation of backlight brightness. In other words, a value of 50(%) does not mean half brightness.

Now back to the fade effect mentioned above. You can, for example, use a `FOR/NEXT` loop and send the increasing brightness value to the PWM pin to create a fade-up effect. To see this in action, type the following (case insensitive) at the command prompt (look at the IPS display when you hit the Enter key):

```
FOR i=0 to 100:PWM 2,800,i,50:PAUSE 10:NEXT i
```

If you get an error, it will be because of a syntax (spelling) error. In this case, simply retype the line, being careful to check that it is as shown above before you press the Enter key.

CLS command

One of the very first BASIC commands that most people learnt back in the 1980s was the **CL**ear **S**creen command, `CLS`. With the circles still on the screen, type `CLS` at the command prompt and watch the screen when you hit Enter. Remember, the graphics commands that we are working through here only affect the attached (IPS) screen, they do not act on the console screen. You will recall from last month's article that it is certain Escape Codes that affect and control the behaviour of the console screen.

In MMBASIC, we can add an optional colour parameter to `CLS`. To do this, we use the `RGB(colourname)` feature, replacing `colourname` with one of the following recognised colours: *`red`*, *`green`*, *`blue`*, *`cyan`*, *`magenta`*, *`yellow`*, *`black`*, *`white`*, *`brown`*, *`gray`*. Now type the following (case insensitive): `CLS RGB(red)`

No prizes for guessing what this will do. Take a few minutes to try the other colours from the list. Note that the whole screen is changed to the colour specified (or currently black if the optional colour parameter is not specified).

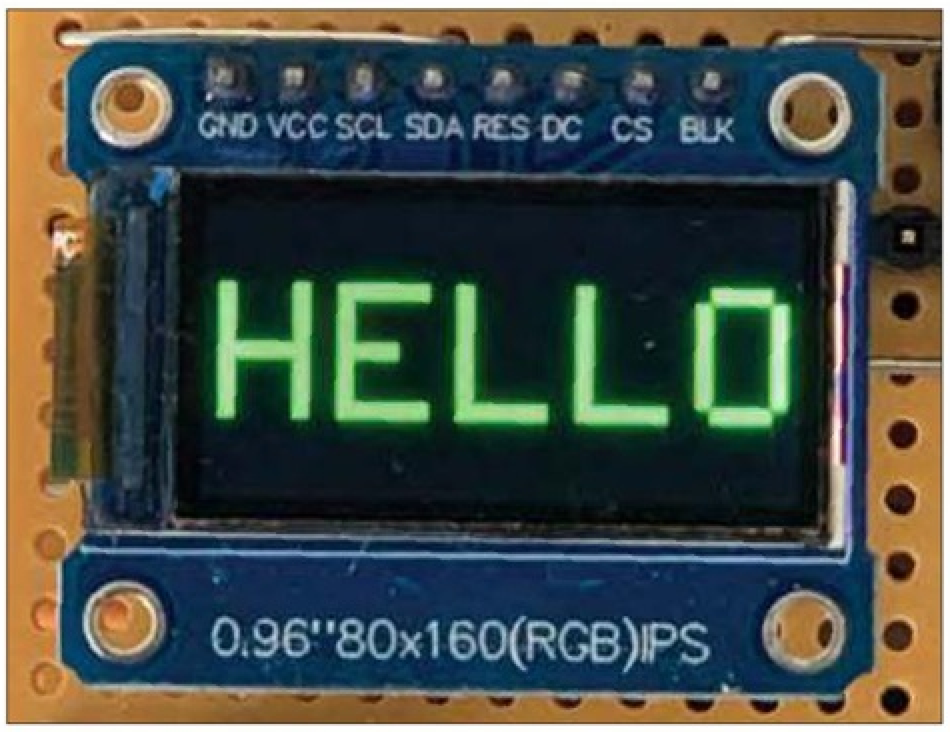


Fig.6. The result of the command: `TEXT 80,40,"HELLO",cm,1,4,rgb(green)`

TEXT command

By now you are reasonably familiar with the `PRINT` command. It is used to display alphanumerics on the console screen. The `TEXT` command is simply the equivalent for displaying alphanumerics on the attached (IPS) screen. However, there are several additional parameters that it needs. The command structure is as follows (do also refer to the *Micromite User Manual*):

```
TEXT x, y, string$ [,alignment] [,font] [,scale]
[,ink_colour] [,paper_colour]
```

It looks complex, but actually it has been designed to be both easy to use and powerful. All the parameters in square brackets are optional (and will take a default value if left unspecified). So to keep things easy for now, just type `CLS` (screen to black), and then type: `TEXT 0,0,"Hello World!"`

You will now see the words printed (in white) in the top left corner of the IPS screen – not very exciting, and certainly not very colourful, but we will be expanding on this.

Screen coordinates

To understand what happened with the last command we first need to talk about screen coordinates. Any point on the screen can be specified by a pair of numbers (the coordinates) called '*x*' and '*y*'. The *x*-coordinate represents the horizontal position (across the screen), and the y-coordinate represents the vertical position . The top-left corner of the screen is defined as coordinate *x* = 0 and *y* = 0. Moving across to the right of the screen means a higher *x* value, and moving down the screen means a higher *y* value. The coordinates MMBASIC use are defined in terms of individual screen pixels.

Our IPS screen has a pixel resolution of 160 pixels wide by 80 pixels high. So with the top-left corner being defined as position *x* = 0, *y* = 0; this means the opposite corner (in the bottom-right) has coordinates *x* = 159, *y* = 79.

The first two parameters of the `TEXT` command always need to be specified as they represent the screen position where the text will appear. And as we have just seen with the previous `TEXT` example, *x* = 0 and *y* = 0 resulted in the text being in the top left corner. Note that most of the other graphical commands also require the *x* and *y* coordinates to be specified in order to define the relevant screen position.

Changing text appearance

To introduce how to use some of the other parameters of the `TEXT` command, we will now repeat the above but use the coordinates for the centre of the screen, which are simply half way along the 160 pixel width, and half way down the 80 pixel height; so this means *x* = 80 and *y* = 40. Now type: `TEXT 80,40,"HELLO"`

Alignment

Notice that the word `HELLO` is not centred in the middle of the screen. This is because by default, the alignment uses the top-left corner of the displayed text to be positioned at the specified coordinates. Therefore, the top-left corner of the letter 'H' is at position *x* = 80, *y* = 40. The alignment parameter allows us to use different parts of the displayed text to be positioned at the specified coordinates. Two letters are used to specify the alignment, the first is the horizontal alignment and can be either: `L`(eft), `C`(entre), or `R`(ight) – similar to left, centre, and right justification in a wordprocessor. The second alignment letter is the vertical alignment and it can be either: `T`(op), `M`(iddle), or `B`(ottom). Note that as with most things in MMBASIC, the alignment letters are not case sensitive.

So, back to our previous example, to get the word `HELLO` centred in the middle of the screen, we need to use the two-letter alignment `CM` (Centre-Middle). Type `CLS` and then: `TEXT 80,40,"HELLO",CM`

This time you will see the word `HELLO` in the middle of the screen. These different alignments are useful in different scenarios. For example, if you want to display a number (of varying length) in the bottom-right hand corner of the screen, then use the `TEXT` command with coordinates *x* =159, and *y* = 79, and with `RB` alignment – try it with: `TEXT 159,79,"12345",rb`

No matter how long the number, it will not run over the right (or bottom) edge of the screen. Take a few minutes to experiment with some of the other alignment combinations.

Font size

The next two parameters, *`font`* and *`scale`* affect the size of the font displayed by the `TEXT` command. The *`font`* parameter is a number and simply defines which font to use – although in the standard Micromite there is just one built-in font (number `1`). It is possible to add user-defined fonts (ie, different styles and different sizes of font) and this topic will be covered next month. For now, just use `1` for the *`font`* parameter. The *`scale`* parameter is a number between 1 and `15` and is the 'pixel multiplier' value that determines the size of the displayed characters. For example, a *`scale`* of `2` will double both the width and the height of the characters. To see this type `CLS` and then: `TEXT 80,40,"HELLO",cm,1,2`

You will now see the word `HELLO` displayed twice as big. Repeat the above `TEXT` command but increase *`scale`* to `3` (there is no need to clear the screen as the bigger font size will overwrite the smaller font), and then repeat with *`scale`* values of `4` and `5`. Notice that *`scale`* `4` just fits on the screen, whereas *`scale`* `5` runs over the edges (but it is still centred in the middle of the screen). For the IPS screen used here with a finite number of pixels, larger *`scale`* values are meaningless and will result in no characters appearing to be displayed (`6` is the highest value we can use in the example here).

Font Colour

Expanding on the above (and using *`scale`* `4`), type `CLS` and then: `TEXT 80,40,"HELLO",cm,1,4,RGB(GREEN)`. This defines the *`ink_colour`* parameter and you will see `HELLO` in green (and filling the screen due to the *`scale`* parameter of `4`) – see Fig.6. Any of the colours previously mentioned can be used, so try some others (vary the *`scale, alignment`*, and *x*, *y* coordinates too).

So far, all the text displayed has been on a black background, but there are situations where we may want a coloured background; eg, to highlight a warning. To do this we use the last parameter – *`paper_colour`*. Type `CLS` and then: `TEXT 80,20," ALARM! ",cm,1,2,rgb(red),rgb(yellow)` – see Fig.7.

You should now have a good understanding of the `TEXT` command, and how to vary the parameters to give many different effects. Refer back to the *IDM* test program and look at how the parameters are being used in each of the six lines of code that use

FD819MX77

Listing 1: The IDM_TestCode program. The unshaded code is the actual program code, and the shaded code is the ‘IPS screen driver’. Note that many lines from the IPS driver have been removed here to allow the listing to fit on the page.

```
` Make it with Micromite
` IPS Display Module (IDM) - Test Code
` Code written: July 2019

` ---------------------------------------------------------------------------------------------------------
` IMPORTANT NOTE:
` ===============
`
` After installing the code - PRESS THE RESET BUTTON on the DM to load the IPS DRIVER. Then type RUN.
`
` If you get any error, power down, then power back up!
` ---------------------------------------------------------------------------------------------------------

` SETUP
` =====
`
Option autorun on                               ` ensure program starts automatically on power-up
IR DevCode, KeyCode, IR_Int                     ` set up an Infra-Red interrupt (to test IR and Piezo)
SetTick 60000,TimeFromRTC                       ` every 60 seconds, update MM time & date from RTC (to avoid drift)
TimeFromRTC                                     ` call SUB to initially get the correct time and date from the RTC
PWM 2,800,100,50                                ` PWM 2 (800Hz): backlight 100% (ON), buzzer 50% (ON when Pin15=0v)
Box 0,0,160,80,3,RGB(green),RGB(blue)           ` clear IPS screen to a blue background with a thin green border
colon                                           ` call SUB to draw a colon for the clock (between hours and minutes)


` MAIN PROGRAM
` ============
`
Do
  Text 31,25,Left$(Time$,2),cm,1,3,RGB(yellow),RGB(Blue)               ` display hours (font 1, scale 3)
  Text 88,25,Mid$(Time$,4,2),cm,1,3,RGB(yellow),RGB(Blue)              ` display minutes (font 1, scale 3)
  Text 132,30,Right$(Time$,2),cm,1,2,RGB(white),RGB(blue)              ` display seconds (font 1, scale 2)
  Text 80,58,Left$(Date$,6)+Right$(Date$,2),cm,1,2,RGB(red),RGB(Blue)  ` display date (font 1, scale 2)
Loop


` SUBROUTINES
` ===========
`
Sub Colon                                       ` SUB to draw a colon for the clock (between hours and minutes)
  Text 60,15,”.”,cm,1,1,RGB(white),RGB(blue)      ` draw ‘top’ dot
  Text 60,29,”.”,cm,1,1,RGB(white),RGB(blue)      ` draw ‘bottom’ dot
End Sub


Sub TimeFromRTC                                 ` SUB to get time from RTC and load MM system Time$ and Date$
  RTC gettime                                     ` (refer to User Manual for further details)
End Sub


Sub IR_Int                                      ` IR interrupt SUB (makes a BEEP sound whenever an IR signal detected)
   SetPin 15,dout                                 ` set pin 15 to 0v i.e. piezo will sound (at 800Hz, at 50% duty)
   Pause 10                                       ` brief pause to allow audible ‘beep’
   SetPin 15,off                                  ` return pin 15 to ‘floating’ i.e. switch piezo OFF
End Sub

` IPS DRIVER                                    ` WARNING: DO NOT CHANGE ANY VALUES or CODE below!!!!
` ==========
`
CSub ST7735S
   00000232
    `spi_write_data
    27BDFFE0 AFBF001C AFB10018 AFB00014 309100FF 3C109D00 8E030090 8E02001C
    8064002C 0040F809 24050006 8E030090 8E02001C 8064002D 0040F809 24050005
    3C02BF80 AC515820 3C03BF80 8C625810 30420080 1040FFFD 3C02BF80 8C425820
    3C029D00 8C430090 8C42001C 8064002D 0040F809 24050006 8FBF001C 8FB10018
    8FB00014 03E00008 27BD0020
    `spi_write_command
    27BDFFE0 AFBF001C AFB10018 AFB00014 309100FF 3C109D00 8E030090 8E02001C
    8064002C 0040F809 24050005 8E030090 8E02001C 8064002D 0040F809 24050005
         .
         .
         .
    AC515830 AC575800 AC5E5840 3C049D00 24841314 0411FD67 00000000 8FBF0074
    8FBE0070 8FB7006C 8FB60068 8FB50064 8FB40060 8FB3005C 8FB20058 8FB10054
    8FB00050 03E00008 27BD0078
    `.rodata
    `startup
    37375453 20533533 76697264 6C207265 6564616F 000A0D64
End CSub

Sub mm.startup                         ` system SUB to ensure IPS driver loaded at Power-Up
  ST7735S 22,23,21, 1,1,-1,-2          ` DCpin, RESETpin, CSpin, orientation (1-4)[, invertdisplay(0,1)], xshift, yshift
End Sub
```

Peter Mather

I want to give credit to Peter Mather, based in the UK. He has written the ‘C’ code screen drivers for all of the different screen types that the Micromite is compatible with. These range from tiny mono OLEDs, to larger colour OLEDs, LCDs with sizes from 1.8 to 9-inches, LED matrices, and also IPS screens like the one we are using here. Peter’s work has allowed Geoff Graham (the brain behind MMBASIC) to focus on the features of each of the graphical commands without worrying about the actual screen hardware attached to the Micromite.

the `TEXT` command. You should now be able to appreciate what is happening in the four lines in the Main Program to display the time and date, and also the two lines in the `colon` subroutine to display the ‘:’ character between the hours and the minutes.

Try writing a short program that creates your own colourful screen of different sized text.

COLOUR command

We have just seen how to display coloured text (potentially on a coloured background) on the IPS screen. The colours used are defined by the *`ink_colour`* and *`paper_colour`* parameters in the `TEXT` command. We have also seen how `CLS` (with no parameters) clears the screen to black. In fact, `CLS` clears the screen to the current *`paper_colour`*, which by default is black. (Note that the default *`ink_colour`* is white.)

Now imagine a scenario where we want to have a program that will use the IPS screen mainly with green text on a blue background. To clear the screen, we would need to use `CLS RGB(BLUE)`, and to write text, we would always need to set the *`ink_colour`* parameter to `RGB(GREEN)`, and always set the *`paper_colour`* parameter to `RGB(BLUE)`. None of these are really an issue, but we could save a bit of typing (especially if we have lots of lines of code using `TEXT`) if we were able to set green and blue as the default ink and paper colours. This is where the `COLOUR` command is useful. Type `CLS` (the screen clears to the default black) and then type:

```
COLOUR rgb(green),rgb(blue)
```

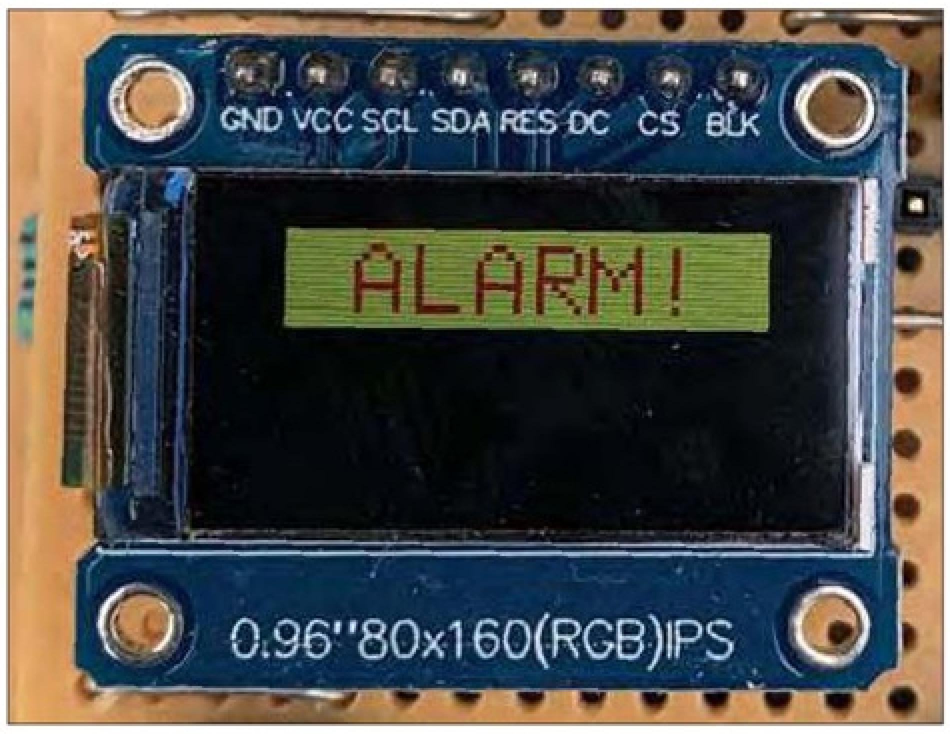


Fig.7. The result of the command: `TEXT 80,20,” ALARM! ”,cm,1,2, rgb(red),rgb(yellow)`

Nothing will appear to have happened; however, this command has now set the default ink and paper colours to green and blue (respectively). Now type `CLS` and see what happens. Then type:

```
TEXT 80,40,”HELLO”,cm,1,3
```

You will notice that the text is now in the default green colour without having to set the *`ink_colour`* parameter in the `TEXT` command. The default colours remain in place until you change them once again with the `COLOUR` command (or when you first power up the Micromite, upon which the colours revert to white (ink) and black (paper)).

More colours please

So far, when we have needed to specify a colour, we have used `RGB(`*`colour_name`*`)` with *`colour_name`* being one of the ten built-in defined colours from the list: *`red`*, *`green`*, *`blue`*, *`cyan`*, *`magenta`*, *`yellow`*, *`black`*, *`white`*, *`brown`*, and *`gray`*. However, it is possible to use many more colours in MMBASIC by using the standard RGB numeric notation that computers use. A computer typically uses a 24-bit number to represent 'any' colour in the spectrum by mixing a certain amount of red, green and blue together. So by using three separate 8-bit binary numbers (three values between 0 and 255 decimal) to represent the three intensity values of red, green, and blue, we are able to define virtually any colour (16,777,216 hues to be precise!). An intensity value of 255 represents full intensity, and a value of 0 represent 'no intensity'. Note: If you have ever used an art, or photo-editing application, on a computer then you will no doubt have seen a colour-selector tool which will highlight the three specific RGB values for any colour selected.

If possible, start an application such as Microsoft Paint, and use the colour selector/editor tool to see the RGB values of any colour you click on. Selecting a pre-defined pale pink colour yields the RGB values as 255,128, and 192 respectively. Once we have the required RGB values, MMBASIC makes it very easy to implement the colour by using the syntax: `RGB(`*`red`*`, `*`green`*`, `*`blue`*`)` where the parameters *`red`*, *`green`* and *`blue`* are simply replaced with the three individual RGB intensity values (between 0 and 255). To see this type `CLS` and then: `TEXT 80,40,”HELLO”,cm,1,4,rgb(255,128,192)`

The result should now be the word `HELLO` in a pale pink colour, filling the entire IPS screen. Experiment further by choosing a different colour that you like. Use either a colour selector tool, or search on the internet for 'RGB colour picker'; and after obtaining the three RGB values, use the `CLS RGB(`*`r`*`,`*`g`*`,`*`b`*`)` command to fill the screen with your chosen colour.

Other graphical commands

Having explained the `TEXT` command in some detail, the following graphical commands will be much easier to follow. The syntax will be shown, along with a brief description, and an example (or two), for each command. It is recommended you try the examples first, and then experiment with each command by changing the parameters to see its effect(s). Use `CLS` when the screen gets too 'busy'; and press the *DM's* reset button if you wish to set the default ink and paper colours back to white and black respectively. Parameters in square brackets are optional.

`PIXEL x,y [,`*`ink_colour`*`]`

Sets the defined pixel at location `x`, `y` to the colour specified (or to the default *`ink_colour`* if no colour is specified).

Example: look very closely, the pixel is small:
`PIXEL 80,40,rgb(magenta)`

`LINE x1,y1,x2,y2 [,`*`line_width`*`] [,`*`ink_colour`*`]`

Draw a straight line between coordinate pairs `x1`,`y1` and `x2`,`y2`. For horizontal or vertical lines (ie, where `x1=x2` or `y1=y2`), the optional *`line_width`* parameter determines the thickness of the line (in pixels) and is a value between 1 and 100. The colour of the line is defined by the optional *`ink_colour`* parameter (or defined by the default *`ink_colour`* if no colour is specified).

Example 1: draw a yellow line three-pixels thick across the top of the screen: `LINE 0,0,159,0,3,rgb(yellow)`

Example 2: draw a diagonal line (in default ink colour) from the bottom-left to the top-right corner. Note that even though a *`line_width`* of `50` is specified, this parameter is ignored (and a value of 1 used) as the line is not purely horizontal or vertical:
`line 0,79,159,0,50`

`BOX x,y,`*`box_width`*`,`*`box_height`*` [,`*`border_width`*`] [,`*`border_colour`*`] [,`*`fill_colour`*`]`

Draw a box with its top-left corner at coordinate `x`,`y`. Parameters *`box_width`* and *`box_height`* determine the size of the box (in pixels). The optional *`border_width`* parameter (defaults to 1) defines the size of the border line that draws the outside edge of the box (and may be set to 0). The optional *`border_colour`* parameter is used to define the colour of the border (or uses default *`ink_colour`* if not specified). The optional *`fill_colour`* parameter will fill the inside of the box with the colour specified. If omitted, the screen area inside the box is left unaffected.

Example 1: draw a blue box with a green border (three pixels thick) over the whole screen area:
`Box 0,0,159,79,3,rgb(0,255,0),rgb(blue)`

Example 2: draw a thick-walled (and unfilled) square in the centre of the screen using the default ink colour:
`BOX 60,20,40,40,10`

`RBOX x,y,`*`box_width`*`,`*`box_height`*` [,`*`border_radius`*`] [,`*`border_colour`*`] [,`*`fill_colour`*`]`

Draws a box with rounded corners. Similar to the above `BOX` command; the top-left corner is at coordinate position `x`,`y`; and the parameters *`box_width`* and *`box_height`* determine the size of the rbox (in pixels). The optional *`border_radius`* parameter (defaults to 10) defines the radius of the rounded corner (in pixels). Note

FD819MX77

that the *`border_width`* is not definable as it is fixed to a width of one pixel. The optional *`border_colour`* parameter is used to define the colour of the border (or uses default *`ink_colour`* if not specified). The optional *`fill_colour`* parameter will fill the inside of the rbox with the colour specified. If omitted, the screen area inside the rbox is left unaffected.

Example 1: draws an ‘athletics track’ shape touching all sides of the screen: `rbox 0,0,159,79,40`

Example 2: draws a white rectangular ‘button’ with a gray border. Note that even though the IPS screen is not a touch screen, the `RBox` example given here is useful for creating on-screen buttons for use with touch-screens. Combine with a `TEXT` command to position a ‘button label’ on top of the button (more on this in a later article when we explore touch-screens):
`RBox 40,30,80,40,10,rgb(gray),rgb(255,255,255)`

`CIRCLE x, y, radius [,border_width] [,aspect_ratio] [,border_colour] [,fill_colour]`
This command draws a circle, centred at position `x,y` and with the specified *`radius`* (in pixels). The optional *`border_width`* parameter is the width of the border used for the circumference and can be zero (it defaults to 1). The optional *`aspect_ratio`* parameter is a (floating point) number and defaults to 1. An *`aspect_ratio`* of 0.5 will draw an oval where the width is half the height. The optional *`border_colour`* parameter is used to define the colour of the border (uses default *`ink_colour`* if not specified). The optional *`fill_colour`* parameter will fill the inside of the circle with the colour specified. If omitted, the screen area inside the circle is left unaffected.

Example 1a: draw a white ‘eye shape’ in the centre of the screen – (continue with next two examples, don’t type `CLS` in between!):
`Circle 80,40,20,1,2,rgb(red),rgb(white)`

Example 1b: draw a green ‘eyeball’:
`Circle 80,40,15,1,1,rgb(gray),rgb(green)`

Example 1c: draw the eye’s pupil:
`Circle 80,40,10,1,0.3, rgb(gray),rgb(black)`

Try experimenting by mixing commands on the screen at the same time (like the last three examples that created an ‘eye’). Mixing text and graphics allows for some professional-looking screens to be created – all under software control.

RTC commands

For completeness, and to understand how to interact with all four hardware elements of the *IDM*, we need to briefly cover MMBASIC’s two RTC commands.

The Micromite has two built-in ‘strings’ (`Time$` and `Date$`) that keep a ‘count’ from whenever the Micromite is powered up – hence, they do not contain the true time and date. We can set them manually, for example: `Time$=“09:43:22″` and `Date$=“21/09/2019″` – however, there are two disadvantages with this. First, we need to manually set the time and date whenever the Micromite is powered up (or is reset); and second, `Time$` ‘drifts’ by a few seconds each hour – so it is not very accurate. Thus, for the Micromite to keep track of the true time, we need to add an RTC module. We saw last month how to physically connect an RTC module. For the software side there are just two simple commands to use: `RTC SETTIME` (to initially set the date and time), and `RTC GETTIME` (to receive time data and automatically set the Micromite’s built-in `Time$` and `Date$`).

Last month, we showed you how to load the correct time into the RTC module by passing six parameters to the `RTC SetTime` command. The syntax is: `RTC SetTime` *`year, month, day, hours, minutes, seconds`*

Providing there is an RTC module attached to the Micromite, and the RTC module has a working back-up battery, then the `RTC SETTIME` command only needs to be used whenever the time (and/or date) needs to be set – the RTC will then accurately keep a track of time. However, this time is ‘inside’ the RTC module and not in the Micromite itself. To get the time information into the Micromite’s built-in `Time$` and `Date$`, the `RTC GetTime` command is used (there are no parameters required for this command). Referring back to the *IDM* test program, you will see the subroutine `SUB TimeFromRTC` contains just a single line of code that uses the `RTC GETTIME` command. So, calling `SUB TimeFromRTC` will load the current ‘correct’ RTC information into `Time$` and `Date$` – however, the Micromite’s `Time$` will still drift, so ideally it needs to be resynchronised with the correct (or at least more accurate) RTC information on a regular basis. This is why in the *IDM* program the `SETTICK` command has been used to set an interrupt that calls `SUB TimeFromRTC` every minute. Now the built-in `Time$` and `Date$` will be kept accurate.

MKC Clock v2

The topics covered this month should make the *IDM* test program much easier to follow; and it can be considered as version ‘v1’ of the *Mini MKC Clock* project. To improve on the software, and also to assist with next month’s topic of user-defined fonts, please download the file: **MKC_Clock_v2.txt** from the October 2019 page of the *PE* website and load it into your *MKC*. However, before you do this, make sure that you have the IPS screen driver (and `sub mm.startup`) installed correctly inside the Library.

On running the *MKC Clock v2* program, everything should look more or less the same as before; however, note that the date is now displayed in a more ‘human-friendly’ format. As usual, the code has comments throughout to explain its operation – but don’t worry if you can’t follow it all. The main thing is to get the code running so that you have a clock displayed correctly (and accurately) on the IPS screen.

Some future ideas for the MKC_Clock

The idea behind the *Mini MKC Clock* is to show how the elements that we are learning about can be implemented into creating something useful. It is important to understand that they are all just ‘building-blocks’, any combination of which may spark some ideas for your own use. We are currently in a position to add functionality to our *Mini MKC Clock* by simply writing the relevant code (and without the need to change the hardware). Here, I just want to highlight some ideas to provoke some thoughts as to what we could add:

- IR Remote – add a menu to select different functions
- Alarm – set an alarm time; optional ‘snooze’ feature
- Timer – set a countdown ‘start’ time; sound the piezo
- Stopwatch – start counting up from 00:00:00
- Thermometer – display temperature with high/low alarms

These features may seem complicated, but it just comes down to coding (as we will see next month…). And this is the real beauty of the Micromite and MMBASIC: it is quick and easy to implement powerful features that would otherwise be difficult.

Next month

We have seen how to use MMBASICs graphical commands. Next month, we will add useful information such as temperature to the *Mini MKC Clock’s* display using an inexpensive DS1820 digital thermometer and a handful of lines of code. We will also add other functionality that allows an IR remote to have overall control of the *Mini MKC Clock*. Plus, to make the time display look more traditional (think seven-segment), we will discuss how to implement user-defined font’s. Until then, have fun!

Max's Cool Beans

By Max the Magnificent

A little more on metastability, and back to the 'HRRG'

There is an old adage that goes: 'There are no stupid questions, only stupid answers.' Of course, this prompts us to ask, 'If there are no stupid questions, then what sort of questions do stupid people ask?'

While you are pondering this conundrum, let me tell you a short story. After university, my first job was in a team at a large company designing central processing units (CPUs) for mainframe computers. Every week, we would all attend an internal lecture on a technical topic. Invariably, the speaker would say something that went right over my head and I would sit there thinking, 'What on earth did that mean?'

I didn't like to raise my hand because I was sure everyone else did understand and they'd think me foolish. Then a more experienced engineer called Joe Taylor would pipe up and say, 'Excuse me, can you explain that point in a little more detail?' And I would breathe a sigh of relief and think, 'Thank you, Joe!'

It was only relatively recently that I came to realise that Joe already knew the answers – he was only asking the questions to help out any inexperienced engineers like yours truly. Now that I'm older and wiser I do the same thing myself; if I'm attending a lecture and I see any younger engineers looking puzzled, I'll put my hand up and ask the question that's on everyone's lips.

Feeling a little unstable

The reason I'm waffling on about not being afraid to ask questions is that I just received an email from a long-time member of the *PE* community querying last month's column saying: 'I really found your last couple of *PE* articles interesting. Your August column really extended my understanding of the SR latch. However, with respect to September's column, I am struggling to see how the 3-clock cycle read of ASYNC (Fig.15) alleviates the metastability problem. Say ASYNC changes state just before t_2 (instead of just before t_1), thereby violating the setup time of DFF1 and causing META to end up in the wrong state due to metastability somewhere between t_2 and t_3. Doesn't this still result in CLEAN being in the incorrect state?'

This is a very good question. To address it, let's remind ourselves that our asynchronous signal from the outside world, which we are assuming is connected to something like a hand-operated toggle switch, is being fed into a two-stage synchroniser (Fig.1).

As we discussed in my previous *Cool Beans* column, when we initiate a read operation in our code, this actually triggers a series of three pulses on the CLK signal. Assuming a transition on the ASYNC signal causes a setup or hold violation on DFF1 on the first active (rising) edge on the CLK signal, then the worst-case scenario is that it will require three active edges on CLK to convey the new value on the ASYNC signal to the CLEAN signal being fed into the CPU. As our astute reader observed, however, we didn't discuss what would happen if the transition on the ASYNC signal were to cause a violation on the second or third active edges on CLK.

The problem is, we've been focusing too closely on our three CLK pulses. Let's 'zoom out' a bit and consider things from a higher level. A typical scenario is that our program will be cycling around a loop in which we perform a bunch of different tasks, one being to read the state of our digital input pin. Let's say we've set our loop to execute once every millisecond. Assuming we're running an Arduino Micro, Uno, or Mega with a 16MHz clock, this means there are 16,000 clock cycles every time we go through our loop, three of which are used to drive our two-stage synchroniser.

To look at this another way, we have a group of three pulses on our CLK signal followed by 15,997 cycles of the main system clock before we see another group of three pulses on our CLK signal. Bearing this in mind, let's consider the various violation possibilities (Fig.2.)

We begin with the ASYNC, META, and CLEAN signals all carrying the same 0 or 1 value (we don't care which), which we've shown in green. At some stage a user flips the toggle switch (see also my column on different types of switches **https://bit.ly/2Mqqk6k**) to its opposite 1 or 0 value, which we've shown in blue.

Let's start with the case of no violations (a). The first edge on CLK loads the new (blue) value on ASYNC into DFF1, appearing on its META output after the register's propagation delay. The second edge on CLK loads the new (blue) value on META into DFF2, appearing on its CLEAN output. So, we're all done after just two CLK cycles (the third CLK simply reloads everything without changing anything).

Now let's consider the transition on ASYNC causing a violation on the first CLK (b), thereby causing DFF1/META to go metastable (pink). Assuming the worst-case scenario, following its recovery time, DFF1 'collapses' into its original, unwanted (green) state. So, it's not until the second edge on CLK that the new value on ASYNC

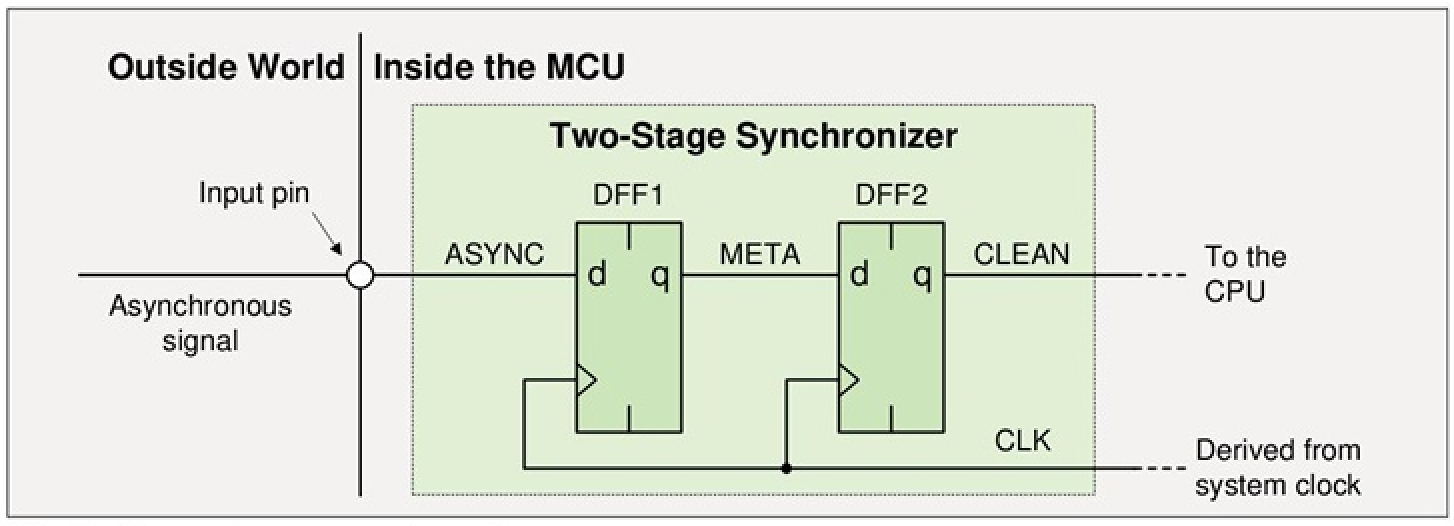


Fig.1. Two-stage synchroniser.

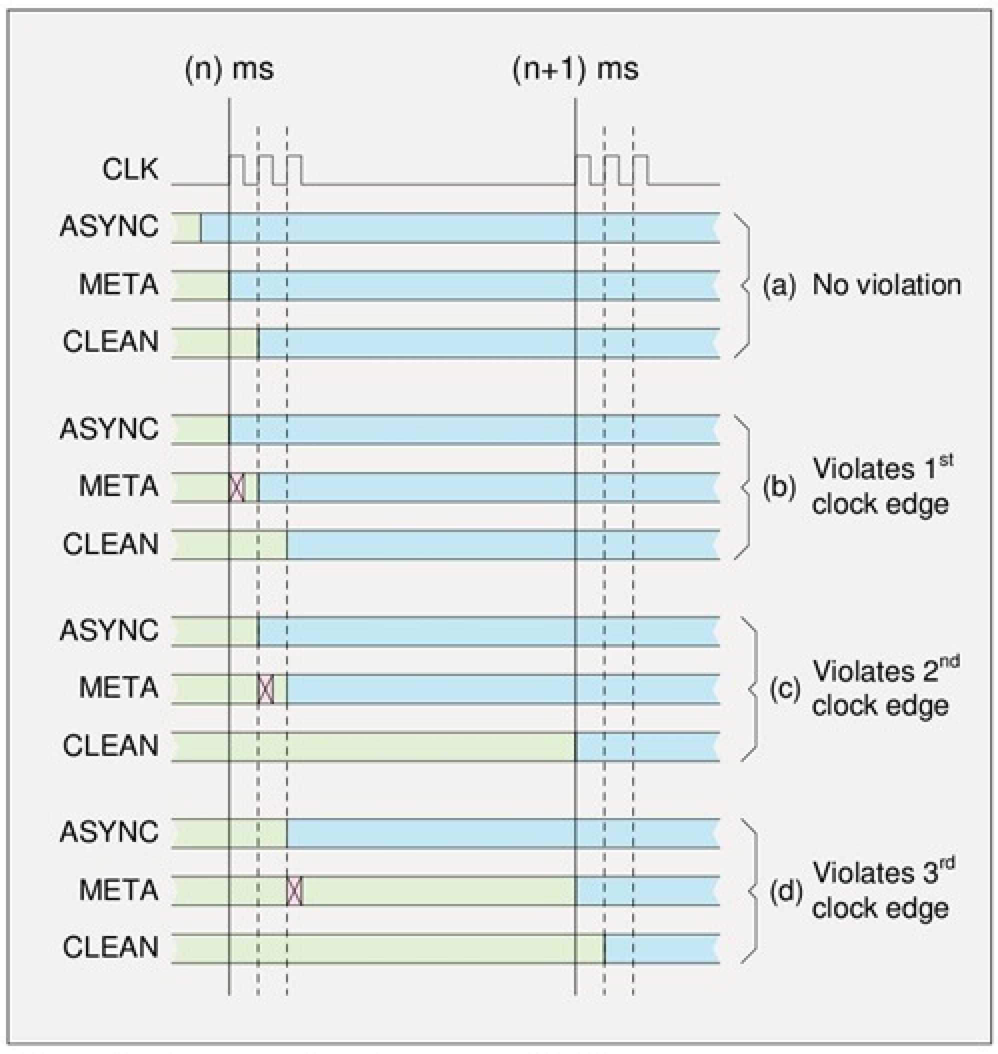


Fig.2. Different violation possibilities.

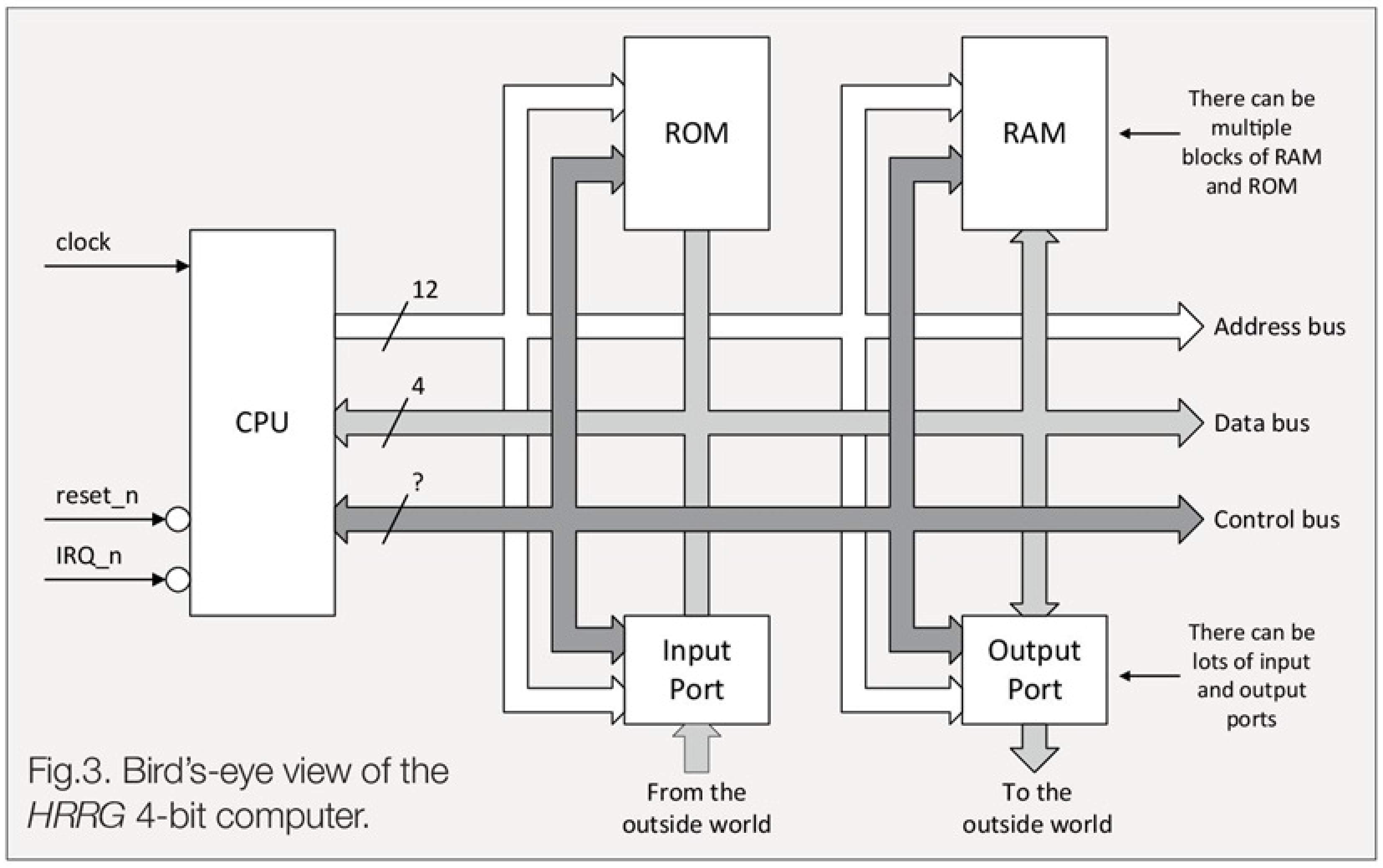


Fig.3. Bird's-eye view of the *HRRG* 4-bit computer.

is loaded into DFF1/META, but this CLK also loads the original, unwanted value on META into DFF2. Thus, it's not until the third edge on CLK that the new, desired value on META is loaded into DFF/CLEAN.

Things start to get interesting when the transition on ASYNC causes violations on the second or third CLK edges, as illustrated in (c) and (d), respectively. The result is that the new value on ASYNC doesn't make its way to the CLEAN signal until the first or second CLK edge in the next group of three pulses one millisecond later.

What we have to realise is that there's a (relatively) long time between transitions on our toggle switch. The absolute fastest we can turn it on and off would be about five times a second, which means that the fastest on-off toggle will take 200ms.

The point of all this is that the two-stage synchroniser's task is to filter out any metastability, which it does admirably. The real worst-case scenarios (violations on the second or third CLK edges in a group) mean that the new signal on the toggle switch isn't seen until the first or second CLK edges in the next group; that is, one millisecond after the switch actually transitions, which is absolutely insignificant in the case of a hand-operated switch.

The 4-bit HRRG Computer

Now, let's return to the *Maxfield-Farr 4-Bit HRRG Computer* (hereafter, just '*HRRG*') discussed in the May 2019 issue of *Practical Electronics*. This is a machine that my co-conspirator, Joe Farr, and I are designing from the ground up. Joe is working on a virtual implementation of this beauty in the form of an emulator that runs on a PC, while I am tasked with defining the physical realisation of the beast. (As an aside, Joe is a member of the EEWeb Experts, **https://bit.ly/2MvmYyS**, and you can discover more about him in a recent column, **https://bit.ly/2GHhGga**).

Currently, I'm working on defining the logical functionality at the level of switches, gates and registers. In the fullness of time, the physical implementation will involve a suite of glass-fronted wooden cabinets, each containing a portion of the *HRRG* realised in a different implementation technology, including relays, vacuum tubes, transistors, and jelly-bean integrated circuits, along with cabinets containing mechanical, magnetic, pneumatic, and hydraulic/fluidic logic.

The idea is that you will be able to start off by creating and running programs in the virtual world on Joe's *HRRG Emulator*, and then gradually add physical cabinets into the mix, where a physical cabinet could contain as little as one 4-bit word of memory or one 4-bit input/output (I/O) port.

One final point before we proceed – in computer terminology, an 8-bit quantity is referred to as a 'byte', while a 4-bit quantity is referred to as a 'nibble' (or 'nybble') on the basis that two nibbles make a byte, thereby showing that engineers do have a sense of humour, albeit not a tremendously sophisticated one.

A birds-eye view

Let's start with a bird's-eye view of the system (Fig.3). This figure is, of course, a simplification – there can be multiple blocs of RAM and ROM and lots of input, output, and bi-directional input/output (I/O) ports. One of the main things to take away from this diagram is that the *HRRG's* data bus is 4-bits (one nibble) wide (as indicated by the line with the '4' annotation).

Also, the address bus is 12-bits (three nibbles) wide, which means we can address 2^{12} = 4096 unique memory locations or I/O ports. We will leave any discussion of the bits forming the control bus for a future column.

Now, although we haven't talked about the *HRRG's* instruction set yet, this does lead to an interesting point, which is the way in which we address our I/O ports. Some processors employ an approach known as port-mapped I/O (PMIO), which involves two sets of instructions like LOAD/STORE to read/write the memory and IN/OUT to read/write the I/O ports. The advantage of this scheme is that the same memory addresses can be used to point to locations in memory or to I/O ports, depending on which instructions are used.

An alternative technique that's simpler to implement and easier to understand is called memory-mapped I/O (MMIO). In this case, the I/O ports share the same address space as the memory, and they are accessed using the same LOAD/STORE instructions. This is the technique we've decided to use in the *HRRG*.

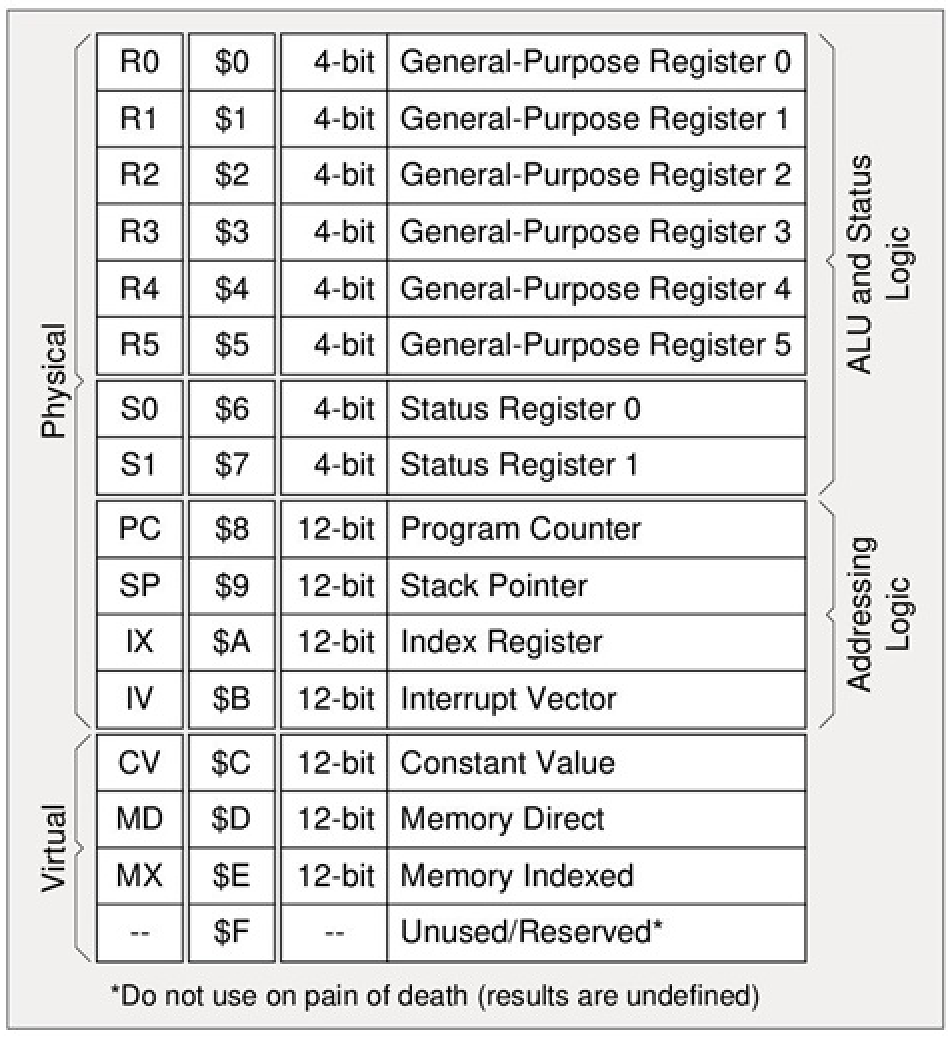
One point I'd like to make while I'm thinking about it is that even though the *HRRG* has only a 4-bit data bus, it is conceptually able to do anything a larger machine can do, just slower and with less efficiency. For example, a 4-bit nibble can only store 2^4 = 16 different values, so we could use it to represent unsigned integers in the range 0 to 15. If we wish to work with larger values, we can simply use more nibbles: two nibbles (8 bits) would allow us to represent unsigned integers in the range 0 to 255, three nibbles (12 bits) would allow us to represent unsigned integers in the range 0 to 4,095, and so on.

The CPU register map

When you are designing a computer, there are myriad decisions and trade-offs to be made, including the number and types of registers you are going to have in the CPU. For example, some of the early microprocessors had a special register called an accumulator (ACC), which was used to store values read from memory and the results of arithmetic and logical operations. Later, some processor designers decided it would be better to have two accumulators (ACC1 and ACC2). Still later, other processor designers opted to provide a bunch of general-purpose registers, each of which could adopt the role of an accumulator.

Since the *HRRG* has a 4-bit data bus that can carry only 2^4 = 16 different values, we decided to have only 16 CPU registers (at least, only 16 registers as far as the user is concerned; there will be additional registers that the CPU uses to perform its magic, but the user doesn't need to know about these). These registers are listed in Fig.4 (the '$' characters indicate hexadecimal values).

As we see, in the case of the data processing and status logic, we decided to have six general-purpose 4-bit registers called R0-R5 and two 4-bit status registers called S0 and S1. Regarding the addressing logic, we opted to have four 12-bit registers to implement the program counter (PC), stack pointer (SP), index register (IX), and interrupt vector (IV). Also, we have three virtual registers called CV (constant value), MD (memory direct), and MX (memory indexed). The way in which all


Physical
Virtual
ALU and Status Logic
Addressing Logic

R0	$0	4-bit	General-Purpose Register 0
R1	$1	4-bit	General-Purpose Register 1
R2	$2	4-bit	General-Purpose Register 2
R3	$3	4-bit	General-Purpose Register 3
R4	$4	4-bit	General-Purpose Register 4
R5	$5	4-bit	General-Purpose Register 5
S0	$6	4-bit	Status Register 0
S1	$7	4-bit	Status Register 1
PC	$8	12-bit	Program Counter
SP	$9	12-bit	Stack Pointer
IX	$A	12-bit	Index Register
IV	$B	12-bit	Interrupt Vector
CV	$C	12-bit	Constant Value
MD	$D	12-bit	Memory Direct
MX	$E	12-bit	Memory Indexed
--	$F	--	Unused/Reserved*

*Do not use on pain of death (results are undefined)

Fig.4. The *HRRG's* CPU register map.

these registers work will become clear as we proceed. Amazingly enough, we ended up with one unused register ($F) which we are reserving for future enhancements.

I think this is a good time to emphasise the fact that there are countless ways of doing everything when it comes to computers. The decisions are made a little trickier when you limit yourself to a 4-bit machine. On the one hand, Joe and I are pretty much making this up as we go along; on the other hand, we are using our decades of experience to 'munge' all sorts of concepts together to come up with something we think is interesting and fun to play with. The point is that nothing is 'written in stone'; if you decide to design your own machine, you might make vastly different decisions.

The status registers

If you ever played with an old 8-bit microprocessor like the 6502, you may recall that it had an 8-bit status register of which three bits remained unused. The five commonly used status bits, or flags, are N (negative), Z (Zero), C (Carry), O (Overflow), and I (Interrupt Mask). In our case, we opted to have two 4-bit status registers (Fig.5.)

The Z flag is primarily used to indicate if the result of a logical or arithmetic operation is zero (a 1 in this flag indicates that all the bits in the result are 0s, while a 0 in the flag indicates that one or more bits in the result contain 1s). Assuming we are working with signed binary numbers, the N flag is primarily used to indicate if the result from an arithmetic operation is negative (a 1 in this flag indicates a negative number; a 0 indicates a positive number). Meanwhile the C and O flags are primarily used to indicate the carry-out and overflow status of a result following an arithmetic operation. (As we will see, the various flags can also be used to indicate other conditions.)

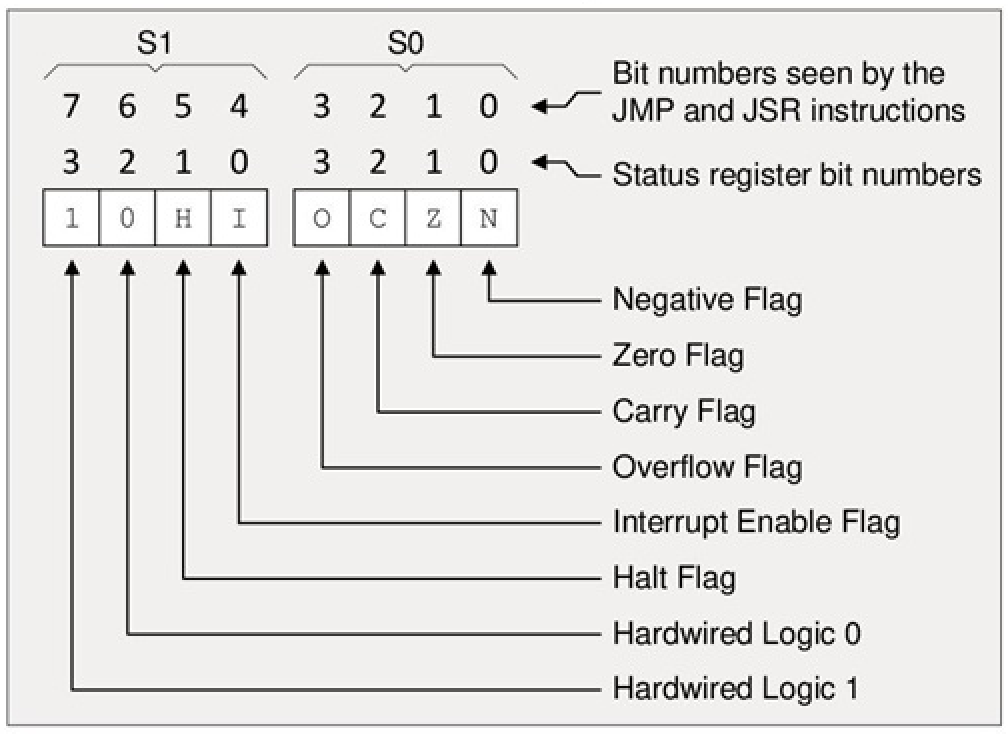


Fig.5. The *HRRG's* status registers and bits/flags.

The I flag is used to enable or disable external interrupts. Following power up or a reset, this flag will contain 0, thereby disabling external interrupts. In order to enable interrupts, this flag must be set to 1 under program control (ie, the currently running program is in charge of setting and clearing this flag).

The H flag is used to halt the CPU. Following power up or a reset, this flag will contain 0, thereby allowing the CPU to run. If the program sets this flag to 1, the CPU will halt. The only way to wake the CPU up again is for an external interrupt to occur (assuming the I flag is set to 1) or for the system to be reset.

The last two flags are hardwired to logic 0 and 1 values. As we will see, these allow us to implement rather cunning JMP (jump) and JSR (jump to subroutine) instructions.

The instruction set

Once again, since the *HRRG* has a 4-bit data bus that can carry only $2^4 = 16$ different values, we decided to have only 16 instructions in our instruction set. This isn't quite as dire as it seems, because each instruction opcode (operation code) nibble is accompanied by one or more operands, which specify the data to be processed (Fig.6).

Don't panic! I know this appears to be complicated, but that's because we are attempting to cram a huge amount of information into a 'one-stop-shop' table. As a quick example of how this all works, let's look at the INC (increment) instruction, whose opcode is $0 (remember '$' indicates a hexadecimal value; we will also use '%' to indicate binary values; eg, %0000). From the table we see that the INC opcode will be followed by a <tar> (target) operand nibble, which will indicate what we wish to increment. If the target is a memory location, then we will have an additional operand <tar-aop> in the form of a 3-nibble memory address.

Suppose we wanted to increment the contents of the 12-bit index register IX. Remember from Fig.4 that the code for this register is $A. So, our 2-nibble machine code would be $0 $A or $0A (opcode = $0 = INC, operand = $A = index register). Alternatively, suppose we wanted to increment the 4-bit memory location at address $462. In this case, we will be using the virtual MD (memory direct) register. From Fig.4 we know that the code for this register is $D. In this case, our 5-nibble machine code would be $0 $D $4 $6 $2, or $0D462 (opcode = $0 = INC, operand = $D = memory direct virtual register, and additional operands $462 to provide the address in memory).

Trust me, that this will all start to make much more sense when we delve deeper in future columns (especially when we start thinking in terms of assembly language). For the moment, just peruse and ponder the following notes associated with the instruction set table:

1. If the source is a 4-bit register or a 12-bit register, then there won't be a <src-aop> (additional operand).
2. If the source is a memory location (as indicated using the MD or MX virtual registers), then the <src-aop> will be a 3-nibble address.
3. If the source is a constant value (indicated using the CV virtual register), then <src-aop> will be a 1-nibble value if the target is a 4-bit register or a memory location; otherwise it will be a 3-nibble value if the target is a 12-bit register.
4. If the target is a 4-bit or 12-bit register, then there won't be a <tar-aop> (additional operand).
5. If the target is a memory location (as indicated using the MD or MX virtual registers), then the <tar-aop> will be a 3-nibble address.
6. If the source is a 4-bit register, the target will typically be a 4-bit register or memory location; if the source is a 4-bit register and the target is one of the 12-bit registers (PC, SP, IX, IV), then the contents of the 4-bit register will be moved (copied) into the least-significant nibble (LSN) of the target register; if the source is a 12-bit register, the target will typically be a 12-bit register or a memory location (see note 7); if the source is a 12-bit register and the target is a 4-bit register, then the contents of the LSN of the 12-bit register will be copied into the 4-bit register.
7. If the contents of a 12-bit register are copied into memory, the target operand will be the least-significant address of a 3-nibble field. If the contents of memory are copied into a 12-bit register, the source operand will be the least-significant address of a 3-nibble field.
8. If the source is a 4-bit register or a memory location, then a 1-nibble value

Fig.6. The *HRRG's* instruction set.

Mnemonic	Opcode	Operands	
INC	$0	<tar>^{4R,12R,Mx} <tar-aop>	Increment contents of register or memory location^{4,5}
DEC	$1	<tar>^{4R,12R,Mx} <tar-aop>	Decrement contents of register or memory location^{4,5}
ADDC	$2	<src>^{CV,4R,Mx} <tar>^{4R,Mx} <src-aop> <tar-aop>	Add contents of <src> and <tar> (with carry); store result in <tar>^{1,2,3,4,5}
SUBB	$3	<src>^{CV,4R,Mx} <tar>^{4R,Mx} <src-aop> <tar-aop>	Subtract contents of <src> from <tar> (with borrow); store result in <tar>^{1,2,3,4,5}
ROLC	$4	<tar>^{4R,Mx} <tar-aop>	Rotate contents of <tar> left through the carry (C) status bit/flag^{4,5}
RORC	$5	<tar>^{4R,Mx} <tar-aop>	Rotate contents of <tar> right through the carry (C) status bit/flag^{4,5}
AND	$6	<src>^{CV,4R,Mx} <tar>^{4R,Mx} <src-aop> <tar-aop>	AND contents of <src> and <tar>; store result in <tar>^{1,2,3,4,5}
OR	$7	<src>^{CV,4R,Mx} <tar>^{4R,Mx} <src-aop> <tar-aop>	OR contents of <src> and <tar>; store result in <tar>^{1,2,3,4,5}
XOR	$8	<src>^{CV,4R,Mx} <tar>^{4R,Mx} <src-aop> <tar-aop>	XOR contents of <src> and <tar>; store result in <tar>^{1,2,3,4,5}
CMP	$9	<src1>^{CV,4R,Mx} <src2>^{4R,Mx} <src1-aop> <src2-aop>	Compare contents of <src> and <tar>; update C and Z status bits/flags
PUSH	$A	<src>^{CV,4R,12R,Mx} <src-aop>	Push contents of <src> onto the stack^{1,2,8,9}
POP	$B	<tar>^{4R,12R,Mx} <tar-aop>	Pop value from stack into <tar>^{4,5,10,11}
JMP	$C	<0/1 #sb> <tar-aop>	Jump to location^{12}
JSR	$D	<0/1 #sb> <tar-aop>	Jump to subroutine^{12}
NOP	$E	--	No operation (don't do anything furiously)
MOV	$F	<src>^{CV,4R,12R,Mx} <tar>^{4R,12R,Mx} <src-aop> <tar-aop>	Move (copy) the contents of <src> (the source) to <tar> (the target)^{1,2,3,4,5,6,7}

<src> = Source
<tar> = Target (destination)
<src-aop> = Additional operand (none if source is a 4R/12R register)
<tar-aop> = Additional operand (none if target is a 4R/12R register)

0/1 = 1-bit logic 0 or 1 value
#sb = 3-bit value specifying status bit to test (0-7)

CV = Constant value
4R = One of the 4-bit registers (R0-R5, S0-S1)
12R = One of the 12-bit registers (PC, SP, IX, IV)
Mx = Either the MD or MX virtual registers.
If MD, the address (operand) is used directly.
If MX, the address (operand) is first added to the contents of the index register (IX)

is pushed onto the stack and SP = SP + 1. If the source is a 12-bit register, then a 3-nibble value will be pushed onto the stack and SP = SP + 3. (The stack pointer will be incremented *after* the PUSH.)

9. If the source is a constant value, then the source operand will be a 1-nibble value, this value will be pushed onto the stack, and SP = SP + 1. (The stack pointer will be incremented *after* the PUSH.)

10. If the target is a 4-bit register or a memory location, then SP = SP – 1 and a 1-nibble value will be popped off the stack (ie, the stack pointer will be decremented *before* the POP). If the target is a 12-bit register, then a 3-nibble value will be popped off the stack (ending up with SP = SP – 3).

11. There are no RTS ('return from subroutine') or RTI ('return from interrupt') instructions – the same effect is achieved by using a POP instruction to retrieve the return address off the stack and load it into the PC.

12. JMP and JSR instructions are followed by a control nibble and then a 3-nibble target address. The control nibble is used to perform unconditional jumps or conditional jumps. The least-significant three bits of the control nibble point to the bit to be tested in the 8-bit status register formed from S1 and S0; this will be in the range 000 to 111 (0 to 7). Assuming a 0 in the most-significant bit of the control nibble, then a value of 1 in the selected status bit will cause a jump; for example, JMP %0001 <target address> is equivalent to a 'Jump if Zero'. By comparison, a 1 in the most-significant bit of the control nibble will invert the operation of the jump; for example, JMP %1001 <target address> is equivalent to a 'Jump if Not Zero'. Note that status bit 7 [bit 3 in S1] is a hard-wired 1, which is equivalent to an unconditional jump.

		Target <tar>							
		4R or Mx				12R			
INC	$0	--	C	Z	N	--	--	Z^7	N^7
DEC	$1	--	C	Z	N	--	--	Z^7	N^7
ADDC	$2	O	C	Z	N	--	--	--	--
SUBB	$3	O	C	Z	N	--	--	--	--
ROLC	$4	--	C^1	Z	N	--	--	--	--
RORC	$5	--	C^2	Z	N	--	--	--	--
AND	$6	--	--	Z	N	--	--	--	--
OR	$7	--	--	Z	N	--	--	--	--
XOR	$8	--	--	Z	N	--	--	--	--
CMP	$9	--	C^4	Z^3	--	--	--	--	--
PUSH	$A	--	--	--	--	--	--	--	--
POP	$B	--	--	Z^5	N^5	--	--	--	--
JMP	$C	--	--	--	--	--	--	--	--
JSR	$D	--	--	--	--	--	--	--	--
NOP	$E	--	--	--	--	--	--	--	--
MOV	$F	--	--	Z^6	N^6	--	--	Z^7	N^7

Fig.7. The instructions and status flags.

Instructions and status flags

Finally, for this column, let's consider the way in which the various instructions affect the status flags as illustrated in Fig.7. In particular, consider the following notes associated with the status flag table:

1. The most-significant bit (MSB) of the target value is copied into the C flag; at the same time, the original contents of the C flag are copied into the least-significant bit (LSB) of the target value.

2. The least-significant bit (LSB) of the target value is copied into the C (carry) flag; at the same time, C flag's original contents are copied into the most-significant bit (MSB) of the target value.

3. The Z flag is set to 1 if the values being compared are equal, otherwise it's cleared to 0.

4. The values being compared are considered to be unsigned integers. The C flag is set to 1 if the value in the <src1> is greater than the value in <src2>; otherwise it's cleared to 0.

5. If the target is status register S1, then its flags will be loaded from the stack and the contents of S0 will not be affected; if the target is S0, then its flags will be loaded from the stack; if the target is any other 4-bit location (register or memory), then the N and Z flags will behave as usual.

6. These flags work as usual for a 4-bit target with one exception – if the target is one of the status registers S0 or S1, then the flags in S0 will not be automatically updated; if the source is a 12-bit register and the target is a 4-bit register, then the Z and N flags will be based on the contents of the least-significant nibble of the 12-bit register (ie, the only bits to be copied); if the source is a 12-bit register and the target is memory, then the Z flag will be set based on the contents of the entire 12-bit register and the N flag will be set based on the contents of the most-significant nibble of the 12-bit register.

7. The Z flag will be set based on the contents of the entire 12-bit register and the N flag will be set based on the contents of the most-significant nibble of the 12-bit register.

Coming soon…

Phew! There's a lot to wrap our brains around here, but things will start to make much more sense when we begin to work in assembly language – the topic of my next column. Until then, have a good one!

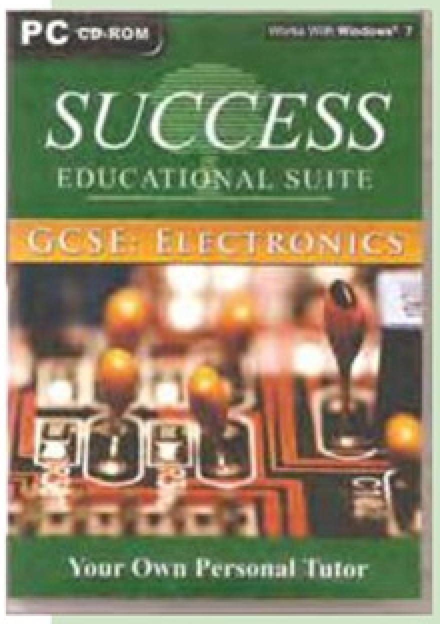
Cool bean Max Maxfield (Hawaiian shirt, on the right) is emperor of all he surveys at **CliveMaxfield.com** – the go-to site for the latest and greatest in technology.

Comments or questions? Email Max at: **max@CliveMaxfield.com**

PRACTICAL ELECTRONICS IS PLEASED TO OFFER YOU THESE

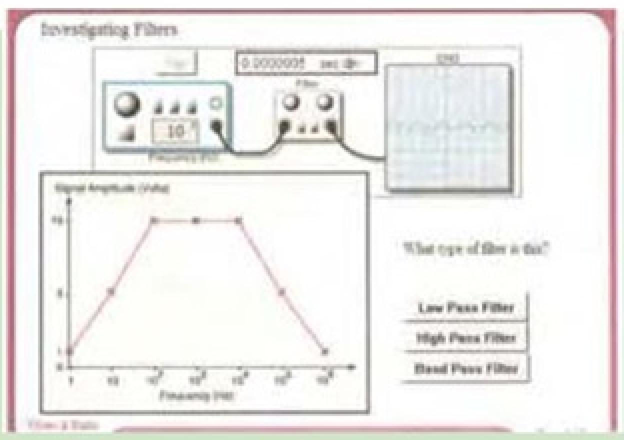
ELECTRONICS CD-ROMS

GCSE ELECTRONICS



£12.50
inc. VAT and P&P

Suitable for any student who is serious about studying and who wants to achieve the best grade possible. Each program's clear, patient and structured delivery will aid understanding of electronics and assist in developing a confident approach to answering GCSE questions. The CD-ROM will be invaluable to anyone studying electronics, not just GCSE students.



*** Contains comprehensive teaching material to cover the National Curriculum syllabus * Regular exercises reinforce the teaching points * Retains student interest with high quality animation and graphics * Stimulates learning through interactive exercises * Provides sample examination ques-tions with model solutions * Authored by practising teachers * Covers all UK examination board syllabuses * Caters for all levels of ability * Useful for self-tuition and revision**

SUBJECTS COVERED

Electric Circuits – Logic Gates – Capacitors & Inductors – Relays – Transistors – Electric Transducers – Operational Amplifiers – Radio Circuits – Test Instruments

Over 100 different sections under the above headings

CIRCUIT WIZARD

Circuit Wizard is a revolutionary software system that combines circuit design, PCB design, simulation and CAD/CAM manufacture in one complete package. Two versions are available – Standard or Professional.

By integrating the entire design process, Circuit Wizard provides you with all the tools necessary to produce an electronics project from start to finish – even including on-screen testing of the PCB prior to construction!

* Circuit diagram design with component library (500 components Standard,1500 components Professional) * Virtual instruments (4 Standard, 7 professional)* On-screen animation * Interactive circuit diagram simulation * True analogue/digital simulation * Simulation of component destruction * PCB Layout * Interactive PCB layout simulation * Automatic PCB routing * Gerber export * Multi-level zoom (25% to 1000%) * Multiple undo and redo * Copy and paste to other software * Multiple document support

This software can be used with the *Jump Start* and *Teach-In 2011* series (and the *Teach-In 4* book).

Standard **£61.25** inc. VAT. Professional **£75** plus VAT.

FD&19MX77

From £49.00

TINA Design Suite V12

Analogue, Digital, Symbolic, RF, MCU and Mixed-Mode Circuit Simulation and PCB Design with TINA

FEATURED IN OUR TEACH-IN 2015 SERIES

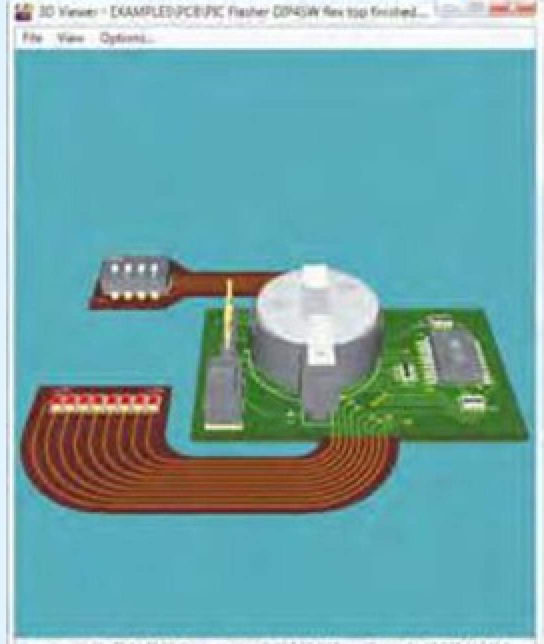
TINA Design Suite V12 is a powerful yet affordable software package for analysing, designing and real time testing analogue, digital, MCU, and mixed electronic circuits and their PCB layouts. You can also analyse RF, communication, optoelectronic circuits, test and debug microcontroller applications.

Enter and analyse any circuit up to 100 nodes (student), or 200 with the Basic (Hobbyist) version within minutes with TINA's easy-to-use schematic editor. Enhance your schematics by adding text and graphics. Choose components from the large library containing more than 10,000 manufacturer models. Analyse your circuit through more than 20 different analysis modes or with 10 high tech virtual instruments.

Present your results in TINA's sophisticated diagram windows, on virtual instruments, or in the live interactive mode where you can even edit your circuit during operation.

Customise presentations using TINA's advanced drawing tools to control text, fonts, axes, line width, colour and layout. You can create and print documents directly inside TINA or cut and paste your results into your favourite word procesing or DTP package.

TINA includes the following Virtual Instruments: Oscilloscope, Function Generator, Multimeter, Signal Analyser/Bode Plotter, Network Analyser, Spectrum Analyser, Logic Analyser, Digital Signal Generator, XY Recorder.

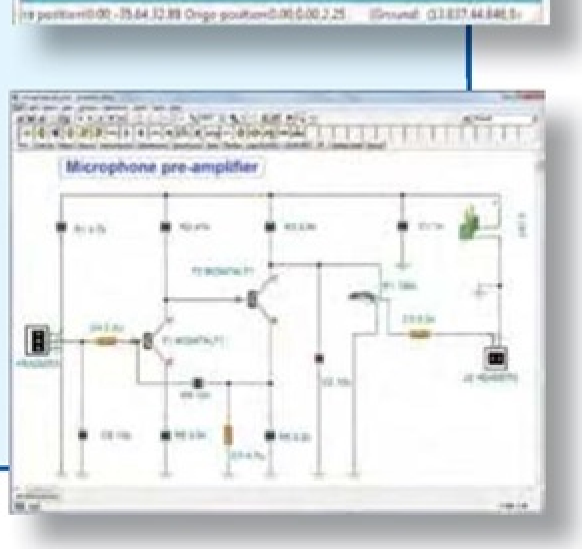


This offer gives you a CD-ROM – the software will need registering (FREE) with Designsoft (TINA), details are given within the package.

Get TINA 12 Design Suite (Hobbyist) for £129 or Student V12 version for £49

Prices include VAT and UK postage

+ get a 1 year free subscription for TINACloud the breakthrough cloud version of TINA which you can run on most operating systems and computers, including PCs, Macs, thin clients iPads and other tablets – even on many smart phones, smart TVs and e-book readers.



To order please either fill out and return the order form, or call us on 01202 880299

Alternatively you can order via our secure online shop at: www.epemag.com

PICmicro TUTORIALS AND PROGRAMMING

HARDWARE

PICmicro Multiprogrammer Board and Development Board

Suitable for use with the three software packages listed below

This flexible PICmicro microcontroller programmer board and combination board allows students and professional engineers to learn how to program PICmicro microcontrollers as well as program a range of 8, 18, 28 and 40 pin devices from the 12, 16 and 18 series PICmicro ranges. For those who want to learn, choose one or all of the packages below to use with the hardware.

- Makes it easier to develop PICmicro projects
- Supports low cost Flash-programmable PICmicro devices
- Fully featured integrated displays – 16 individual LEDs, quad 7-segment display and alphanumeric LCD display
- Supports PICmicro microcontrollers with A/D converters
- Fully protected expansion bus for project work
- USB programmable
- Compatible with the E-blocks range of accessories

£159.99 including VAT and postage (worldwide)

SOFTWARE

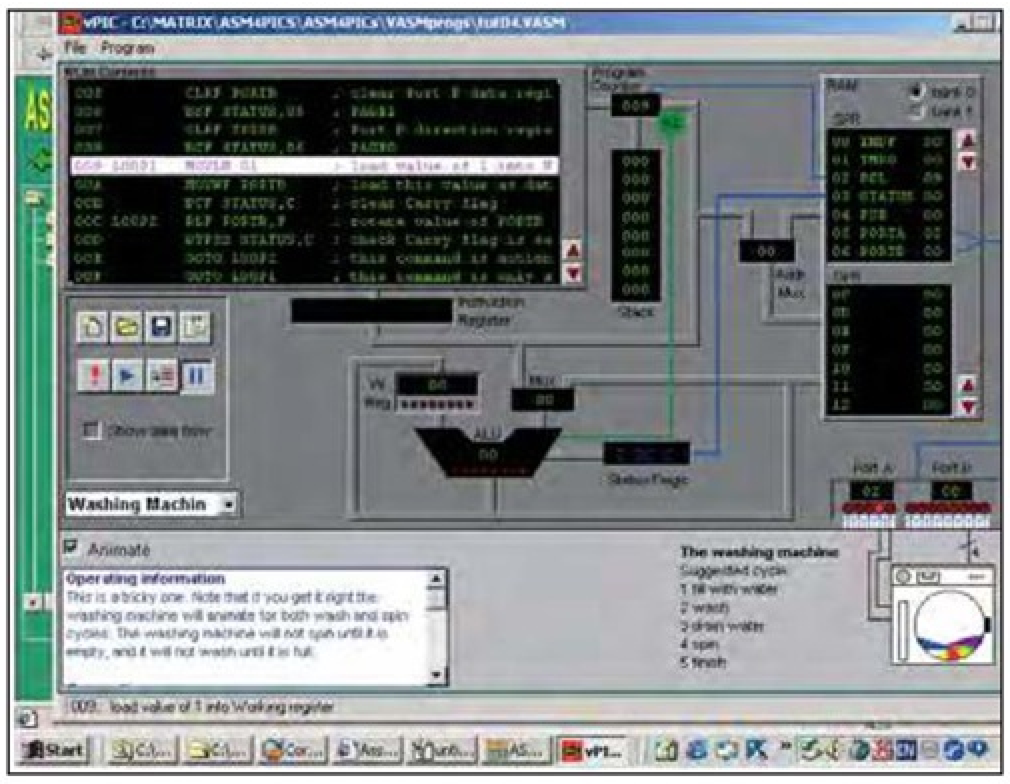
ASSEMBLY FOR PICmicro V7

(Formerly PICtutor)

Assembly for PICmicro microcontrollers V7.0 (previously known as PICtutor) by John Becker contains a complete course in programming the PIC16F84, 16F88 and 16F877a PICmicro microcontroller from Arizona Microchip. It starts with fundamental concepts and extends up to complex programs including watchdog timers, interrupts and sleep modes.

The CD makes use of the latest simulation techniques which provide a superb tool for learning: the Virtual PICmicro microcontroller, this is a simulation tool that allows users to write and execute MPASM assembler code for the PIC16F84 microcontroller on-screen. Using this you can actually see what happens inside the PICmicro MCU as each instruction is executed, which enhances understanding.

- Comprehensive instruction through 45 tutorial sections
- Includes Vlab, a Virtual PICmicro microcontroller: a fully functioning simulator
- Tests, exercises and projects covering a wide range of PICmicro MCU applications
- Includes MPLAB assembler
- Visual representation of a PICmicro showing architecture and functions
- Expert system for code entry helps first time users
- Shows data flow and fetch execute cycle and has challenges (washing machine, lift, crossroads etc.)
- Imports MPASM files.

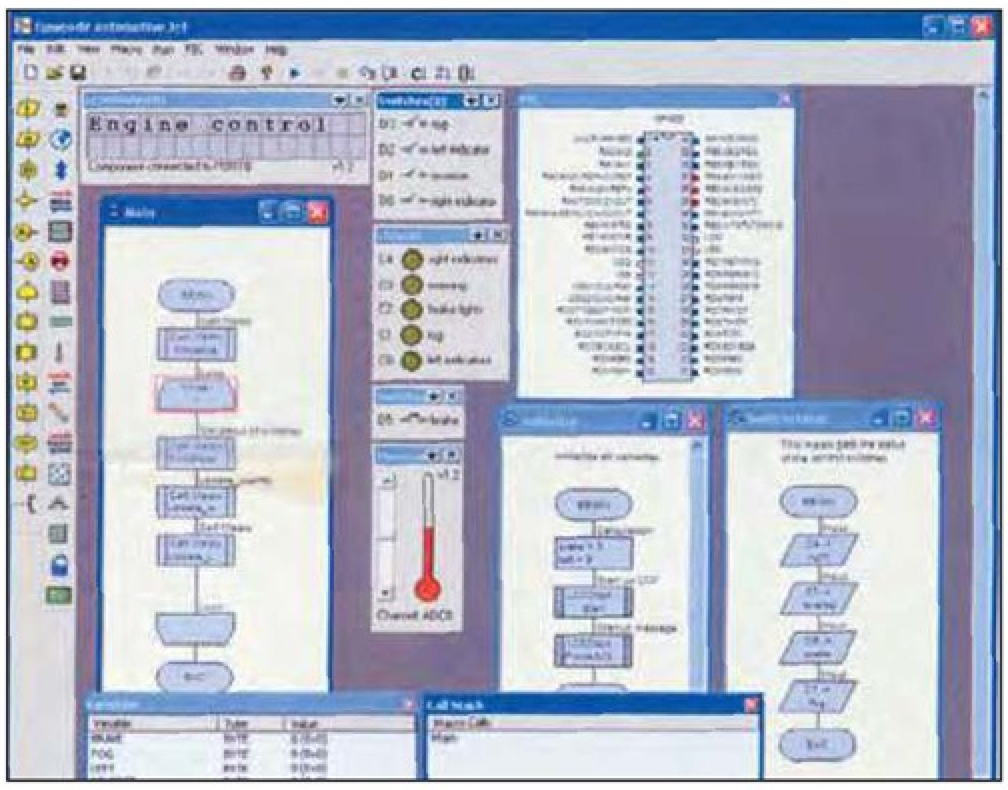


FLOWCODE FOR PICmicro V8

Flowcode is a very high level language programming system based on flowcharts. Flowcode allows you to design and simulate complex systems in a matter of minutes. A powerful language that uses macros to facilitate the control of devices like 7-segment displays, motor controllers and LCDs. The use of macros allows you to control these devices without getting bogged down in understanding the programming. When used in conjunction with the development board this provides a seamless solution that allows you to program chips in minutes.

- Requires no programming experience
- Allows complex PICmicro applications to be designed quickly
- Uses international standard flow chart symbols
- Full on-screen simulation allows debugging and speeds up the development process.
- Facilitates learning via a full suite of demonstration tutorials
- Produces code for a wide range of devices
- 16-bit arithmetic strings and string manipulation
- Pulse width modulation
- I2C.

Please note: Due to popular demand, Flowcode is now available as a download. Please include your email address and a username (of your choice) on your order. A unique download code will then be emailed to you.



PRICES

Prices for each of the CD-ROMs above are:

(Order form on next page)

(UK and EU customers add VAT to 'plus VAT' prices)

Single License . . . . £99 plus VAT

Site Licence . . . . £499 plus VAT

Flowcode . . . . Contact us for pricing
(choose PIC-8b, PIC-16b, PIC-32b, AVR/Arduino,ARM)

ELECTRONICS TEACH-IN 2

ELECTRONICS TEACH-IN 2 CD-ROM
USING PIC MICROCONTROLLERS A PRACTICAL INTRODUCTION

This *Teach-In* series of articles was originally published in *EPE* in 2008 and, following demand from readers, has been collected together in the *Electronics Teach-In 2* CD-ROM.

The series is aimed at those using PIC microcontrollers for the first time. Each part of the series includes breadboard layouts to aid understanding and a simple programmer project is provided.

Also included are 29 *PIC N' Mix* articles, also republished from *EPE*. These provide a host of practical programming and interfacing information, mainly for those that have already got to grips with using PIC microcontrollers. An extra four part beginners guide to using the C programing language for PIC microcontrollers is also included.

The CD-ROM also contains all of the software for the *Teach-In 2* series and *PIC N' Mix* articles, plus a range of items from Microchip – the manufacturers of the PIC microcontrollers. The material has been compiled by Wimborne Publishing Ltd. with the assistance of Microchip Technology Inc.

CD-ROM **Order code ETI2 CD-ROM** **£9.50**

ELECTRONICS TEACH-IN 3

ELECTRONICS TEACH-IN 3 CD-ROM

The three sections of this CD-ROM cover a very wide range of subjects that will interest everyone involved in electronics, from hobbyists and students to professionals. The first 80-odd pages of Teach-In 3 are dedicated to *Circuit Surgery*, the regular *EPE* clinic dealing with readers' queries on circuit design problems – from voltage regulation to using SPICE circuit simulation software.

The second section – *Practically Speaking* – covers the practical aspects of electronics construction. Again, a whole range of subjects, from soldering to avoiding problems with static electricity and indentifying components, are covered. Finally, our collection of *Ingenuity Unlimited* circuits provides over 40 circuit designs submitted by the readers of *EPE*.

The CD-ROM also contains the complete *Electronics Teach-In 1* book, which provides a broad-based introduction to electronics in PDF form, plus interactive quizzes to test your knowledge, TINA circuit simulation software (a limited version – plus a specially written TINA Tutorial).

The Teach-In 1 series covers everything from Electric Current through to Microprocessors and Microcontrollers and each part includes demonstration circuits to build on breadboards or to simulate on your PC.

CD-ROM **Order code ETI3 CD-ROM** **£8.50**

ELECTRONICS TEACH-IN 4

ELECTRONICS TEACH-IN 4 CD-ROM
A BROAD-BASED INTRODUCTION TO ELECTRONICS.

The Teach-In 4 CD-ROM covers three of the most important electronics units that are currently studied in many schools and colleges. These include, Edexcel BTEC level 2 awards and the electronics units of the Diploma in Engineering, Level 2.

The CD-ROM also contains the full Modern Electronics Manual, worth £29.95. The Manual contains over 800 pages of electronics theory, projects, data, assembly instructions and web links.

A package of exceptional value that will appeal to all those interested in learning about electronics or brushing up on their theory, be they hobbyists, students or professionals.

CD-ROM **Order code ETI4 CD-ROM** **£8.99**

ELECTRONICS TEACH-IN BUNDLE – FOR PARTS 3, 4 & 5

CD-ROM **Order code ETIB3 CD-ROM** **£18.95**

ORDERING
ALL PRICES INCLUDE UK POSTAGE

Standard/Student/Basic (Hobbyist) Version price includes postage to most countries in the world EU residents outside the UK add £5 for airmail postage per order

ELECTRONICS TEACH-IN 5

FREE CD-ROM
ELECTRONICS TEACH-IN 5
£8.99
FROM THE PUBLISHERS OF EPE
JUMP START
PRACTICALLY SPEAKING
The techniques of project construction
PIC 'N MIX
Starting out with PIC microcontrollers
FREE CD-ROM
TEACH-IN 2
TWO TEACH-INs FOR THE PRICE OF ONE!
Electronics Teach-In 2

ELECTRONICS TEACH-IN 5
JUMP START

FREE CD-ROM

15 design and build circuit projects dedicated to newcomers or those following courses in school and colleges. The projects are: Moisture Detector, Quiz Machine, Battery Voltage Checker, Solar-Powered Charger, Versatile Theft Alarm, Spooky Circuits, Frost Alarm, Mini Christmas Lights, iPod Speaker, Logic Probe, DC Motor Controller, Egg Timer, Signal Injector Probe, Simple Radio Receiver, Temperature Alarm.

PLUS:

PIC' N MIX – starting out with PIC Microcontrollers and PRACTICALLY SPEAKING – the techniques of project construction.

FREE CD-ROM – The free CD-ROM is the complete Teach-In 2 book providing a practical introduction to PIC Microprocessors plus MikroElektronika, Microchip and L-Tek PoScope software.

160 Pages **Order code ETI5** **£8.99**

Please send me: **CD-ROM ORDER FORM**

Version required:

- **☐ Assembly for PICmicro V6**
- **☐ 'C' for 16 Series PICmicro V6**

- **☐ Single Licence**
- **☐ Site licence**

Note: The software on each version is the same, only the licence for use varies.

☐ PICmicro Multiprogrammer Board and Development Board (hardware)

- **☐ Circuit Wizard – Standard**
- **☐ Circuit Wizard – Professional**
- **☐ GCSE Electronics**
- **☐ TINA Design Suite V11 Basic (Hobbyist)**
- **☐ TINA Design Suite V11 (Student)**

- **☐ Teach-In 2**
- **☐ Teach-In 3**
- **☐ Teach-In 4**
- **☐ Teach-In Bundle (3, 4 and 5)**

Full name . . . .

Address . . . .

. . . . Post code . . . . Tel . . . .

Email: . . . .

☐ I enclose cheque/PO in £ sterling payable to Practical Electronics for £ . . . .

☐ Please charge my Visa/Mastercard: £ . . . .

Valid From: . . . . Card expiry date: . . . .

Card No: . . . .

Card Security Code . . . .
(The last 3 digits on or just under the signature strip)

Single License and Site License Versions – overseas readers add £5 to the basic price of each order for airmail postage (**do not add VAT** unless you live in an EU (European Union) country, then add VAT at 20% or provide your official VAT registration number).

Send your order to:
Wimborne Publishing Ltd
113 Lynwood Drive
Wimborne
Dorset BH21 1UU

To order by phone call:

01202 880299. Fax: 01202 843233
Goods are normally sent within seven days

E-mail: stewart.kearn@wimborne.co.uk

Online shop:

www.epemag.com

FD&19MX77

DIRECT BOOK SERVICE

Teach-In 2017

Introducing the BBC micro:bit

SH1
0
1
2
3V
GND

For Beginners of Any Age
Python Coding on the BBC micro:bit
Jim Gatenby

PYTHON CODING ON THE BBC MICRO:BIT

Jim Gatenby

Python is the leading programming language, easy to learn and widely used by professional programmers. This book uses MicroPython, a version of Python adapted for the BBC Micro:bit.

Among the many topics covered are: main features of the BBC micro:bit including a simulation in a web browser screen; various levels of programming languages; Mu Editor for writing, saving and retrieving programs, with sample programs and practice exercises; REPL, an interactive program for quickly testing lines of code; scrolling messages, creating and animating images on the micro:bit's LEDs; playing and creating music, sounds and synthesized speech; using the on-board accelerometer to detect movement of the micro:bit on three axes; glossary of computing terms.

This book is written using plain English, avoids technical jargon wherever possible and covers many of the coding instructions and methods which are common to most programming languages. It should be helpful to beginners of any age, whether planning a career in computing or writing code as an enjoyable hobby.

118 Pages **Order code PYTH MBIT** £7.99

For Beginners of Any Age
Getting Started with the BBC micro:bit
Mike Tooley

GETTING STARTED WITH THE BBC MICRO:BIT

Mike Tooley

Not just an educational resource for teaching youngsters coding, the BBC micro:bit is a tiny low cost, low-profile ARM-based single-board computer. The board measures 43mm × 52mm but despite its diminutive footprint it has all the features of a fully fledged microcontroller together with a simple LED matrix display, two buttons, an accelerometer and a magnetometer.

Mike Tooley's book will show you how the micro:bit can be used in a wide range of applications from simple domestic gadgets to more complex control systems such as those used for lighting, central heating and security applications. Using Microsoft Code Blocks, the book provides a progressive introduction to coding as well as interfacing with sensors and transducers.

Each chapter concludes with a simple practical project that puts into practice what the reader has learned. The featured projects include an electronic direction finder, frost alarm, reaction tester, battery checker, thermostatic controller and a passive infrared (PIR) security alarm.

No previous coding experience is assumed, making this book ideal for complete beginners as well as those with some previous knowledge. Self-test questions are provided at the end of each chapter, together with answers at the end of the book. So whatever your starting point, this book will take you further along the road to developing and coding your own real-world applications.

108 Pages **Order code BBC MBIT** £7.99

THEORY AND REFERENCE

PRACTICAL ELECTRONICS HANDBOOK – 6th Ed
Ian Sinclair
440 pages **Order code NE21** £33.99

STARTING ELECTRONICS – 4th Ed
Keith Brindley
296 pages **Order code ELSEV100** £18.99

ELECTRONIC CIRCUITS – FUNDAMENTALS & APPLICATIONS – Updated version
Mike Tooley
400 pages **Order code TF43** £32.99

FUNDAMENTAL ELECTRICAL AND ELECTRONIC PRINCIPLES – 3rd Ed
C.R. Robertson
368 pages **Order code TF47** £21.99

A BEGINNER'S GUIDE TO TTL DIGITAL ICs
Robert Penfold
142 pages **OUT OF PRINT BP332** £5.45

UNDERSTANDING ELECTRONIC CONTROL SYSTEMS
Owen Bishop
228 pages **Order code NE35** £36.99

The books listed here have been selected by the *Practical Electronics* editorial staff as being of special interest to everyone involved in electronics and computing. They are supplied by mail order direct to your door.

FOR A FULL DESCRIPTION OF THESE BOOKS AND CD-ROMS SEE THE SHOP ON OUR WEBSITE

www.epemag.com

All prices include UK postage

MICROPROCESSORS

INTERFACING PIC MICROCONTROLLERS – 2nd Ed
Martin Bates
298 pages **Order code NE48** £34.99

PROGRAMMING 16-BIT PIC MICROCONTROLLERS IN C – LEARNING TO FLY THE PIC24
Lucio Di Jasio (Application Segments Manager, Microchip, USA)
496 pages + CD-ROM **Order code NE45** £38.00

INTRODUCTION TO MICROPROCESSORS AND MICROCONTROLLERS – 2nd Ed
John Crisp
222 pages **Order code NE31** £29.99

THE PIC MICROCONTROLLER YOUR PERSONAL INTRODUCTORY COURSE – 3rd Ed
John Morton
270 pages **Order code NE36** £25.00

PIC IN PRACTICE – 2nd Ed
David W. Smith
308 pages **Order code NE39** £24.99

MICROCONTROLLER COOKBOOK
Mike James
240 pages **Order code NE26** £36.99

BOOK ORDERING DETAILS

All prices include UK postage.

For airmail, add £3 per book to Europe, £4 for rest of the world per book.

CD-ROM prices include VAT and/or postage to anywhere in the world.

Send a cheque, (£ sterling only) made payable to: **Practical Electronics** or credit card details (Visa or Mastercard) to:

WIMBORNE PUBLISHING LIMITED,
113 LYNWOOD DRIVE, WIMBORNE, DORSET BH21 1UU

Books are normally sent within seven days of receipt of order.

Please check price (see latest issue of Practical Electronics or website) before ordering from old lists.

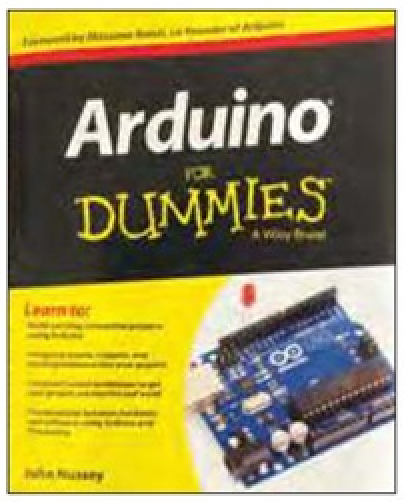
For a full description of these books please see the shop on our website.
Tel: 01202 880299 – E-mail: stewart.kearn@wimborne.co.uk

Order from our online shop at: www.epemag.com

ARDUINO

Teach-In 2016

See opposite for our popular introduction to the Arduino

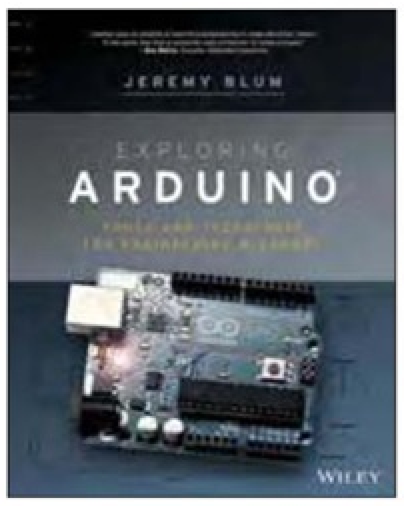


ARDUINO FOR DUMMIES
John Nussey

Arduino is no ordinary circuit board. Whether you're an artist, a designer, a programmer, or a hobbyist, Arduino lets you learn about and play with electronics. You'll discover how to build a variety of circuits that can sense or control real-world objects, prototype your own product, and even create interactive artwork. This handy guide is exactly what you need to build your own Arduino project – what you make is up to you!

- **Learn by doing** – start building circuits and programming your Arduino with a few easy examples – right away!
- **Easy does it** – work through Arduino sketches line by line, and learn how they work and how to write your own.
- **Solder on!** – don't know a soldering iron from a curling iron? No problem! You'll learn the basics and be prototyping in no time.
- **Kitted out** – discover new and interesting hardware to turn your Arduino into anything from a mobile phone to a Geiger counter.
- **Become an Arduino savant** – find out about functions, arrays, libraries, shields and other tools that let you take your Arduino project to the next level
- **Get social** – teach your Arduino to communicate with software running on a computer to link the physical world with the virtual world

438 Pages Order code ARDDUM01 £19.99



EXPLORING ARDUINO
Jeremy Blum

Arduino can take you anywhere. This book is the roadmap.

Exploring Arduino shows how to use the world's most popular microcontroller to create cool, practical, artistic and educational projects. Through lessons in electrical engineering, programming and human-computer interaction, this book walks you through specific, increasingly complex projects, all the while providing best practices that you can apply to your own projects once you've mastered these. You'll acquire valuable skills – and have a whole lot of fun.

- **Explore** the features of commonly used Arduino boards
- **Use** Arduino to control simple tasks or complex electronics
- **Learn** principles of system design, programming and electrical engineering
- **Discover** code snippets, best practices and system schematics you can apply to your original projects
- **Master** skills you can use for engineering endeavours in other fields and with different platforms

357 Pages Order code EXPARD01 £26.99

COMPUTING AND ROBOTICS

NEWNES INTERFACING COMPANION
Tony Fischer-Cripps
295 pages Order code NE38 £41.00

HOW TO BUILD A COMPUTER MADE EASY
Robert Penfold
120 pages Order code BP707 £8.49

EASY PC CASE MODDING
Robert Penfold
192 pages + CD-ROM Order code BP542 £8.99

FREE DOWNLOADS TO PEP-UP AND PROTECT YOUR PC
Robert Penfold
128 pages Order code BP722 £7.99

WINDOWS XP EXPLAINED
N. Kantaris and P.R.M. Oliver
264 pages Order code BP514 £7.99

THE INTERNET – TWEAKS, TIPS AND TRICKS
Robert Penfold
128 pages Order code BP721 £7.99

eBAY – TWEAKS, TIPS AND TRICKS
Robert Penfold
128 pages Order code BP716 £7.50

AN INTRODUCTION TO eBAY FOR THE OLDER GENERATION
Cherry Nixon
120 pages Order code BP709 £8.49

HOW TO FIX YOUR PC PROBLEMS
Robert Penfold
128 pages Order code BP705 £8.49

AN INTRODUCTION TO WINDOWS VISTA
P.R.M. Oliver and N. Kantaris
120 pages Order code BP703 £8.49

WINDOWS 8.1 EXPLAINED
180 Pages Order code BP747 £10.99

AN INTRODUCTION TO THE NEXUS 7
118 Pages Order code BP744 £8.99

COMPUTING FOR THE OLDER GENERATION
Jim Gatenby
308 pages Order code BP601 £8.99

ANDROIDS, ROBOTS AND ANIMATRONS
Second Edition – John Iovine
224 pages Order code MGH1 £16.99

ROBOT BUILDERS COOKBOOK
Owen Bishop
366 pages Order code NE46 £26.00

INTRODUCING ROBOTICS WITH LEGO MINDSTORMS
Robert Penfold
288 pages + Order code BP901 £14.99

MORE ADVANCED ROBOTICS WITH LEGO MINDSTORMS
Robert Penfold
298 pages Order code BP902 £14.99

HOW TO FIX YOUR PC PROBLEMS
Robert Penfold
128 pages Order code BP705 £8.49

WINDOWS 7 – TWEAKS, TIPS AND TRICKS
Andrew Edney
120 pages Order code BP708 £8.49

GETTING STARTED IN COMPUTING FOR THE OLDER GENERATION
Jim Gatenby
120 pages Order code BP704 £8.49

WINDOWS 8.1 EXPLAINED
Noel Kantaris
180 Pages Order code BP747 £10.99

COMPUTING WITH A LAPTOP FOR THE OLDER GENERATION
Robert Penfold
120 pages Order code BP702 £8.49

AN INTRODUCTION TO EXCEL SPREADSHEETS
Jim Gatenby
18 pages Order code BP701 £8.49

KINDLE FIRE HDX EXPLAINED
118 Pages Order code BP743 £8.99

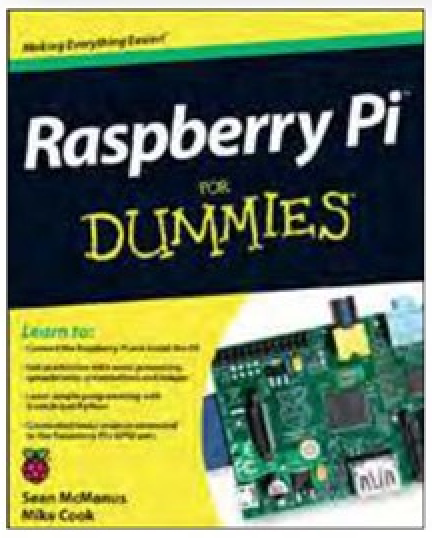
FD&19MX77

AUDIO & VIDEO

VALVE AMPLIFIERS – 4th Ed
Morgan Jones
288 pages Order code ELSEV33 £46.99

BUILDING VALVE AMPLIFIERS
Morgan Jones
368 pages Order code NE40 £29.00

RASPBERRY PI



RASPBERRY Pi FOR DUMMIES
Sean McManus and Mike Cook

Write games, compose and play music, even explore electronics – it's easy as Pi! The Raspberry Pi offers a plateful of opportunities, and this great resource guides you step-by-step, from downloading, copying, and installing the software to learning about Linux and finding cool new programs for work, photo editing, and music. You'll discover how to write your own Raspberry Pi programs, create fun games, and much more!

Open this book and find:
What you can do with Python; Ways to use the Raspberry Pi as a productivity tool; How to surf the web and manage files; Secrets of Sonic Pi music programming; A guide to creating animations and arcade games; Fun electronic games you can build; How to build a 3D maze in Minecraft; How to play music and videos on your Raspberry Pi.

400 Pages Order code RPiDUM01 £17.99

RASPBERRY Pi MANUAL: A practical guide to the revolutionary small computer
176 pages Order code H001 £17.99

RASPBERRY Pi USER-GUIDE – 4th Ed
262 pages Order code JW001 £20.90

PROGRAMMING THE RASPBERRY Pi
192 pages Order code MGH4 £10.99

GETTING STARTED WITH RASPBERRY Pi
164 pages Order code OR01 £11.50

TEACH-IN BOOKS

ELECTRONICS TEACH-IN 6

ONLY AVAILABLE ON CD-ROM

ELECTRONICS TEACH-IN 6
A COMPREHENSIVE GUIDE TO RASPBERRY Pi

Mike & Richard Tooley

Teach-In 6 contains an exciting series of articles that provides a complete introduction to the Raspberry Pi, the low-cost computer that has taken the education and computing world by storm.

This latest book in our Teach-In series will appeal to electronic enthusiasts and computer buffs wanting to get to grips with the Raspberry Pi.

Anyone considering what to do with their Pi, or maybe they have an idea for a project but don't know how to turn it into reality, will find Teach-In 6 invaluable. It covers: Programming, Hardware, Communications, Pi Projects, Pi Class, Python Quickstart, Pi World, and Home Baking.

The CD-ROM also contains all the necessary software for the series so that readers can get started quickly and easily with the projects and ideas covered.

160 Pages **Order code ETI6** **£8.99**

ELECTRONICS TEACH-IN 7 (Includes free CD-ROM)

RUNNING LOW – ORDER NOW!

ELECTRONICS TEACH-IN 7
DISCRETE LINEAR CIRCUIT DESIGN

Mike & Richard Tooley

Teach-In 7 is a complete introduction to the design of analogue electronic circuits. Ideal for everyone interested in electronics as a hobby and for those studying technology at schools and colleges. Supplied with a free cover-mounted CD-ROM containing all the circuit software for the course, plus demo CAD software for use with the Teach-In series

Discrete Linear Circuit Design* Understand linear circuit design* Learn with 'TINA' – modern CAD software* Design simple, but elegant circuits* Five projects to build: Pre-amp, Headphone Amp, Tone Control, VU-meter, High Performance Audio Power Amp.

PLUS

Audio Out – an analogue expert's take on specialist circuits
Practically Speaking – the techniques of project building

160 Pages **Order code ETI7** **£8.99**

ELECTRONICS TEACH-IN 8

ONLY AVAILABLE ON CD-ROM

ELECTRONICS TEACH-IN 8
INTRODUCING THE ARDUINO

Mike & Richard Tooley

Hardware – learn about components and circuits; Programming – powerful integrated development system; Microcontrollers – understand control operations; Communications – connect to PCs and other Arduinos

This exciting series has been designed for electronics enthusiasts who want to get to grips with the inexpensive, immensely popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. Teach-In 8 will provide a one-stop source of ideas and practical information.

The Arduino offers a remarkably effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle through wireless or the Internet. Teach-In 8 is based around a series of practical projects with plenty of information to customise each project.

This book also includes *PIC n' Mix*: PICs and the PICkit 3 - A Beginners guide by Mike O'Keefe and *Circuit Surgery* by Ian Bell - State Machines part 1 and 2.

The CD-ROM includes files for Teach-In 8 plus Microchip MPLAB IDE XC8 8-bit Compiler and PICkit 3 User Guide. Also included is Lab-Nation Smartscope software.

160 Pages **Order code ETI8** **£8.99**

VISIT OUR WEBSITE FOR MORE BOOKS AND FAST, EASY ONLINE ORDERING
WWW.EPEMAG.COM

BOOK ORDER FORM

Full name: ....

Address: ....

....

.... Post code: .... Telephone No: ....

Email: ....

☐ I enclose cheque/PO payable to Practical Electronics for £ ....

☐ Please charge my card £ .... Card expiry date..

Card Number ....

Valid From Date .... Card Security Code .... (The last three digits on or just below the signature strip)

Please send book order codes: ....

....

....

Please continue on separate sheet of paper if necessary

THE BASIC SOLDERING GUIDE

LEARN TO SOLDER SUCCESSFULLY!
ALAN WINSTANLEY

The No.1 resource for learning all the basic aspects of electronics soldering by hand.

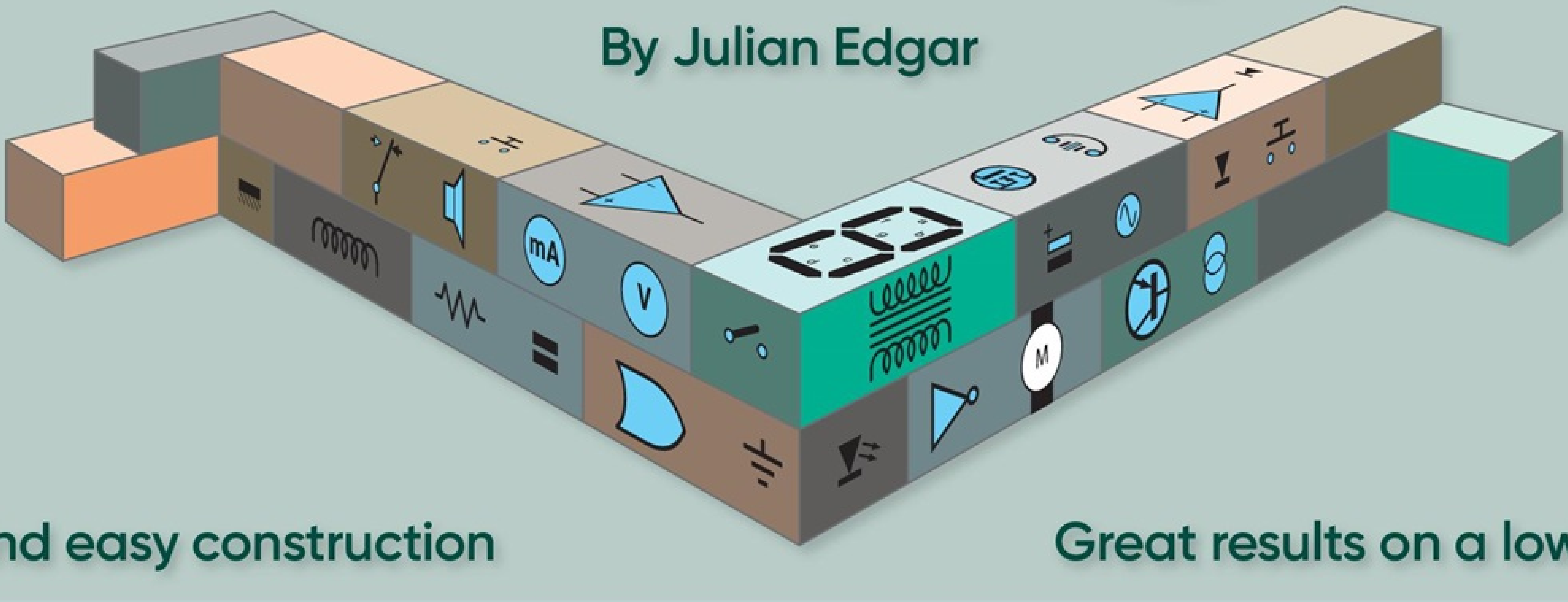
With more than 80 high quality colour photographs, this book explains the correct choice of soldering irons, solder, fluxes and tools. The techniques of how to solder and desolder electronic components are then explained in a clear, friendly and non-technical fashion so you'll be soldering successfully in next to no time! The book also includes sections on reflow soldering and desoldering techniques, potential hazards, useful resources and a very useful troubleshooting guide.

Also ideal for those approaching electronics from other industries, the *Basic Soldering Guide Handbook* is the best resource of its type, and thanks to its excellent colour photography and crystal clear text, the art of soldering can now be learned by everyone!

86 Pages **Order code AW1** **£9.99**

Electronic Building Blocks

By Julian Edgar



Quick and easy construction

Great results on a low budget

Automotive gadgetry

This month, I am going to look at a variety of items that are suitable for use with your car. However, even if you're not interested in automotive electronics, all the products also have non-automotive uses.

12V battery monitor with USB power

This small module is designed to work with a lead-acid 12V battery (eg, in a car), but it could be added to any equipment that provides nominally 12V power. It combines a numerical display of battery supply voltage, an 8-LED bar graph that also shows battery voltage, and two USB 5V power supply outputs. Designed to clip into a 46 × 26mm hole, the module flush-mounts like a normal panel meter. The USB outputs are in the front face of the module, allowing easy access when the module is installed.

The device works down to about 6V – by 'working', I mean that the USB outputs stay at 5V and the display operates. At that voltage (which of course would indicate a very sick 12V battery) the bar graph is flashing a solitary red LED; this indicator of low battery voltage occurs at any voltage less than 10V. The first green LED on the bar graph lights with an input of 11V, and the last LED at 13.3V. The USB output under no-load conditions is about 5.2V; this drops to 5.1V under a 1A load.

The digital display was reasonably accurate – to within 0.1V – and at 1A load, the module didn't get warm. However, like many very low-cost modules, it might be wise to de-rate it a bit, rather than trying to pull the claimed 2A.

Cost is just under £3 delivered. Search on eBay under '12V Lead Acid Battery BMS Capacity Level Indicator Volt Meter Dual USB Charger' (at the time of writing, eBay item 253777635432). In addition to installation in a car, the module is ideal for equipment like 12V power supplies.

FD&19MX7Z

LED brake light flasher

In some countries, the centre high-mount LED brake light in cars is configured to flash a small number of pulses when first activated. This draws the attention of the following driver to the fact that you're decelerating. After the initial pulses, the brake light then stays on until the brake pedal is released. The module shown here is easily added to an LED brake light to achieve the same pulsing outcome.

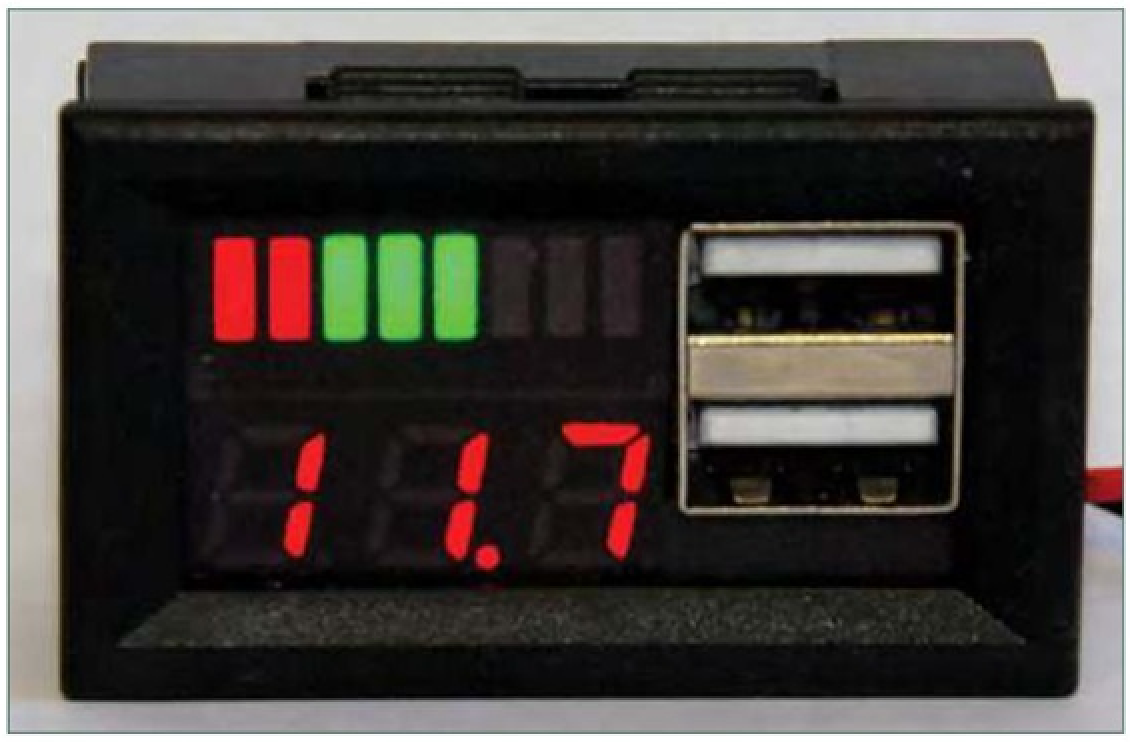


Fig.1. This low-cost 12V module integrates bar graph and digital battery voltage displays with dual USB power outputs.

Only about 50 × 30 × 14mm, the module is housed in a box that is small enough to be tucked almost anywhere. The electronics are also potted, so the module should be largely weatherproof. Wiring is straightforward: red and black 'in' connections, and similarly polarised 'out' connections. In use, the wires to the LED brake light are cut and then the module inserted – very easy.

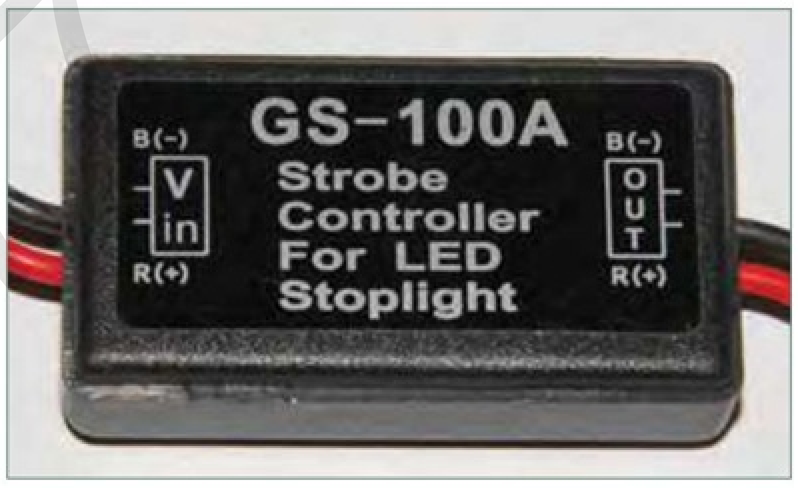


Fig.2. On braking, this module will flash a centre-high-mount LED brake light. After some initial flashes to attract the attention of the following driver, the brake light resumes its steady-on mode.

Claimed power handling is 15W (plenty for any LED brake light) and cost is low at £4 delivered – to find it, search on eBay under 'UK 12V-24V Flash Strobe Controller Flasher Module for LED Brake Stop Light Lamp' (at the time of writing, eBay item 222685223667).

In use, the controller causes the brake light to flash very fast for about a second, then pulses more slowly for about 3s, before staying fully illuminated until the brake is released.

In addition to the brake light function, the module is ideal anywhere a warning needs to be given that equipment has been switched on or is moving.

Note: you must check the legality of flashing brake lights in your jurisdiction before fitting this module to a vehicle.

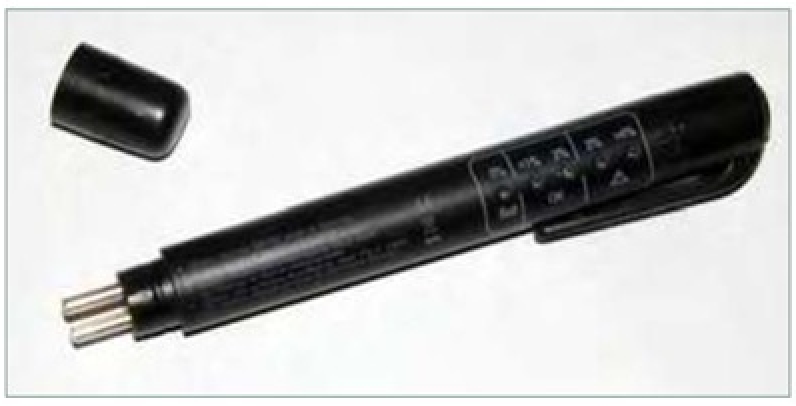


Fig.3. Shaped and sized like a pen, this handy device measures the percentage of water in your car's brake fluid. If the water content is too high, get the fluid changed.

Brake fluid water detector

One thing that few people know is that brake fluid in cars is hygroscopic – it absorbs water from the atmosphere. That's a primary reason why car manufacturers have fixed service periods for brake fluid – if it is not changed at appropriate intervals, its moisture content rises. So why is that unacceptable? Corrosion of internal brake components can then more readily occur.

So that's where this gadget comes in. It comprises a pen-shaped instrument that has five LEDs along its length. Behind a removable end-tip are two metal probes. The instrument is switched on (a green LED first shows that the single AAA cell is OK) and then the probes are submerged in the brake fluid – easily accessible at the master cylinder reservoir. Four LEDs then show the percentage of water in the fluid – less than 1%, 2%, 3%, or more than 4%. 2% or less is OK; higher than that is not.

Some testing with new brake fluid showed it had zero water in it, but adding even a tiny quantity of water quickly caused the reading to rise.

You can get the tester for around £3 delivered from Banggood, ID: 1077231.

Magnet-controlled flashing light

I was initially a fraction reluctant to include these lights here, but they work so well that I felt I had to! So what are they? They're small (29mm diameter × 9mm depth) lights that use an included CR2032 button cell. Two lights are supplied in each packet. Each of the lights contains three bright surface-mount LEDs – red in the case of the ones I bought. When triggered, the LEDs flash rapidly for about 8s, before reverting to a lower current consumption mode of three quick flashes about every second. But the trick is in the phrase 'when triggered' – the lights are triggered by the removal of a magnet.

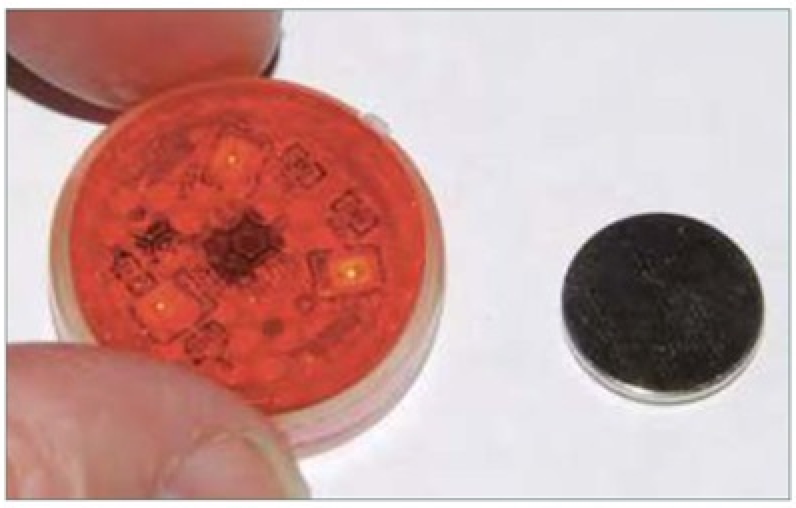


Fig.4. At left is the small flashing LED light, complete with internal button cell. On the right is the magnet, the removal of which triggers the light. The light is designed to mount on the inner edge of a car door, warning approaching drivers when the door is open. When the door is closed, the presence of the magnet switches off the light.

The lights are designed to be placed on the trailing edge of the car door (the part that is exposed when the door is open). When the doors are open, the lights flash a warning to drivers approaching from behind. When the doors are closed, the lights are switched off by the presence of the small supplied magnet. The circular magnet (only a few millimetres thick) comes with a self-adhesive pad and needs to be placed within about 5mm of the light to disable it. The lights also come with double-sided adhesive pads to allow them to be held in place. Of course, you need to ensure that within the car's body design there is room for the light and its magnet (a combined thickness of about 11mm).

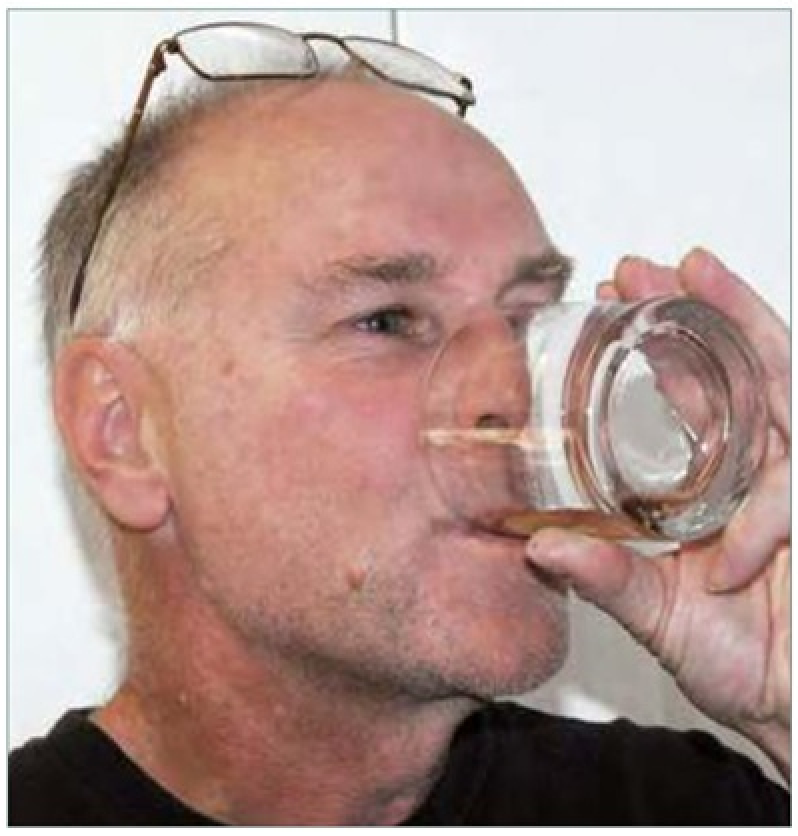
You can find the lights for £3 a pair delivered from Banggood, ID: 1417651.

And the use of the lights isn't limited to just cars. The lights can be used anywhere a warning that something is open is required. I used one on a large kitchen pantry door – a door that's easy to walk into when it's open.

Cheap breathalyser

Finally, we have a low-cost (£5 delivered) alcohol breath tester, available from Banggood, ID: 987311. It's powered by two AAA cells (and an AG13 button cell for the clock), and doesn't have any claimed accuracy – as we will see in a moment, that's just as well!

This cheap breathalyser is fun to play with, but testing it…



... involved consuming standard drinks of whisky (*the things I do for you!*) which did *not* give results that matched my weight and the consumed alcohol.

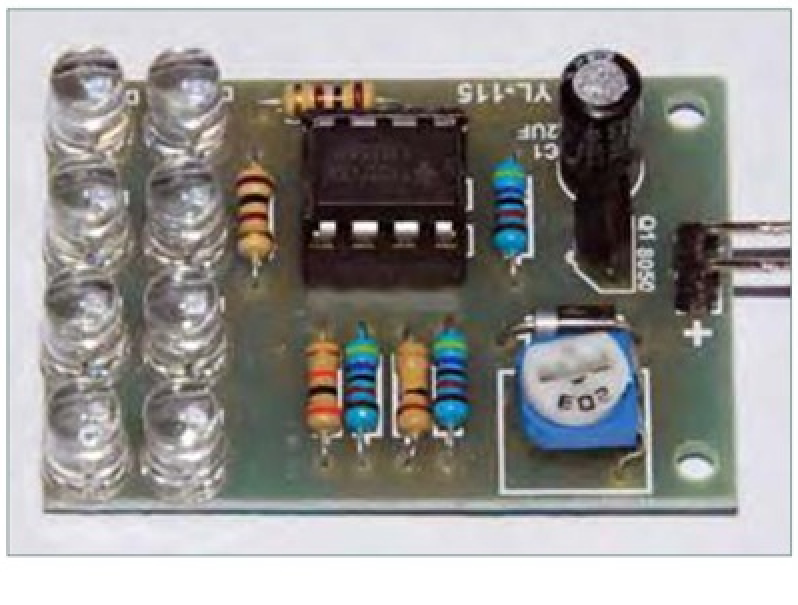
In operation, the device is switched on, whereupon a timer counts down from ten to zero as the sensor warms up. The user then blows into an orifice, after which the blood alcohol level is shown on the digital display as a percentage.

Wanting to do all this scientifically, I rapidly consumed two standard drinks of whisky, that according to reputable online calculators, with my weight, should have given me a blood alcohol level of 0.004. The tester showed 0.003. After 4-6 hours, this should have returned to zero, but the tester didn't show that either. The next evening, I started from scratch, but this time rapidly drinking three standard whiskies. Again, the device showed only 0.003, but by blowing very hard for as long as I could exhale, I got it momentarily up to 0.004 – again, lower than it should have been. However, one positive for the tester is that when I had drunk no alcohol at all, the device always showed a zero reading.

So perhaps it's worthwhile fun for a party, but *definitely not* something to use to evaluate whether or not you are over the legal limit.

Next month

In the next issue, I'll cover four cheap modules based around LEDs that are small and simple kits you build yourself.



Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
DECEMBER 2017		
Precision Voltage and Current Reference – Part 2	04110161	£15.35
JANUARY 2018		
High-Power DC Motor Speed Controller – Part 1	11112161	£12.88
Build the SC200 Amplifier Module	01108161	£12.88
FEBRUARY 2018		
GPS-Syncronised Analogue Clock Driver	04202171	£12.88
High-Power DC Motor Speed Controller – Part 2		
+ Control Board	11112161	£12.88
+ Power Board	11112162	£15.30
MARCH 2018		
Stationmaster Main Board	09103171	£17.75
+ Controller Board	09103172	
Build the SC200 Amplifier Module – Power Supply	01109111	£16.45
APRIL 2018		
Spring Reverberation Unit	01104171	£15.30
DDS Sig Gen Lid	Black	£8.05
DDS Sig Gen Lid	Blue	£7.05
DDS Sig Gen Lid	Clear	£8.05
MAY 2018		
High Performance RF Prescaler	04112162	£10.45
Micromite BackPack V2	07104171	£10.45
Microbridge	24104171	£5.60
JUNE 2018		
High Performance 10-Octave Stereo Graphic Equaliser	01105171	£15.30
JULY 2018		
Touchscreen Appliance Energy Meter – Part 1	04116061	£17.75
Automotive Sensor Modifier	05111161	£12.88
AUGUST 2018		
Universal Temperature Alarm	03105161	£7.05
Power Supply For Battery-Operated Valve Radios	18108171	£27.50
	18108172	
	18108173	
	18108174	
SEPTEMBER 2018		
3-Way Active Crossover	01108171	£22.60
Ultra-low-voltage Mini LED Flasher	16110161	£5.60
OCTOBER 2018		
6GHz+ Touchscreen Frequency Counter	04110171	£12.88
Two 230VAC MainsTimers	10108161	£12.88
	10108162	
NOVEMBER 2018		
Super-7 AM Radio Receiver	06111171	£27.50
FEBRUARY 2019		
1.5kW Induction Motor Speed Controller	10105122	£35.00
MARCH 2019		
10-LED Bargraph Main Board	04101181	£11.25
+Processing Board	04101182	£8.60
APRIL 2019		
Heater Controller	10104181	£14.00
MAY 2019		
2× 12V Battery Balancer	14106181	£5.60
Deluxe Frequency Switch	05104181	£10.45
USB Port Protector	07105181	£5.60

PROJECT	CODE	PRICE
JUNE 2019		
Arduino-based LC Meter	04106181	£8.00
USB Flexitimer	19106181	£10.45
JULY 2019		
Full-wave 10A Universal Motor Speed Controller	10102181	£12.90
Recurring Event Reminder	19107181	£8.00
Temperature Switch Mk2	05105181	£10.45
AUGUST 2019		
Brainwave Monitor	25108181	£12.90
Super Digital Sound Effects Module	01107181	£5.60
Watchdog Alarm	03107181	£8.00
PE Theremin (three boards: pitch, volume, VCA)	PETX0819	£19.50
PE Theremin component pack (see p.56, August 2019)	PETY0819	£15.00
OCTOBER 2019		
Programmable GPS-synced Frequency Reference	04107181	£11.50
Digital Command Control Programmer for Decoders	09107181	£8.75
Opto-isolated Mains Relay (main board)	10107181	£11.50
Opto-isolated Mains Relay (2 × terminal extension board)	10107182	

FD&19MX77

PE/EPE PCB SERVICE

Order Code	Project	Quantity	Price

Name ....

Address ....

....

Tel ....

Email ....

I enclose payment of £ .... (cheque/PO in **£ sterling only**)

payable to: **Practical Electronics**

Card No ....

Valid From .... Expiry Date ....

Card Security No ....

You can also order PCBs by phone, Fax, Email or via the shop on our website on a secure server: www.epemag.com

PCBs for most recent *PE*/*EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, plated-through hole, with solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

We do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.

Back issues of articles are available – see Back Issues page for details.

All prices include VAT and UK p&p. Add £2 per board for airmail outside of Europe. Orders and payment should be sent to:

Practical Electronics, Wimborne Publishing Ltd
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU
Tel 01202 880299 Fax 01202 843233
Email: stewart.kearn@epemag.wimborne.co.uk
On-line Shop: www.epemag.com

Cheques should be made payable to Practical Electronics **(Payment in £ sterling only)**.

NOTE: While 95% of our boards are held in stock and are dispatched within seven days of receipt of order, please allow a maximum of 28 days for delivery if we need to restock.

CLASSIFIED ADVERTISING

Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price our classified page offers excellent value. The rate for semi-display space is £10 (+VAT) per centimetre high, with a minimum height of 2·5cm. All semi-display adverts have a width of 5.5cm. The prepaid rate for classified adverts is 40p (+VAT) per word (minimum 12 words).

Cheques are made payable to Practical Electronics. **VAT must be added**. Advertisements with remittance should be sent to: Practical Electronics, 113 Lynwood Drive, Wimborne, Dorset, BH21 1UU.

Tel 01202 880299 Email: stewart.kearn@wimborne.co.uk

For rates and further information on display and classified advertising please contact our Advertisement Manager, Stewart Kearn – see below.

Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.

We have been the leading monthly magazine in this market for the last twenty-seven years.

BOWOOD ELECTRONICS LTD

Suppliers of Electronic Components

www.bowood-electronics.co.uk

Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield, Derbyshire S40 2QR. Sales: 01246 200 222

Send large letter stamp for Catalogue

The British Amateur Electronic Club at:

baec.tripod.com

Has many interesting articles on computers; digital electronics and analogue electronics.

COAST ELECTRONICS

BREAKOUTS-COMPONENTS-CONTRACT DESIGN-3D PRINTER PARTS-MUSICAL-MICROCONTROLLERS

WWW.COASTELECTRONICS.CO.UK

MISCELLANEOUS

VALVES AND ALLIED COMPONENTS IN STOCK. Phone for free list. Valves, books and magazines wanted. Geoff Davies (Radio), tel. 01788 574774.

PIC DEVELOPMENT KITS, DTMF kits and modules, CTCSS Encoder and Decoder/Display kits. Visit www.cstech.co.uk

Electrical Industries Charity (EIC)

We help people working in the electrical, electronics and energy community as well as their family members and retirees.

We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.

www.electricalcharity.org

Andrew Kenny – Qualified Patent Agent

EPO UKIPO USPTO

Circuits Electric Machinery Mechatronics

Web: www.akennypatentm.com
Email: Enquiries@akennypatentm.com
Tel: 0789 606 9725

ADVERTISING INDEX

BRUNNING SOFTWARE	39
CRICKLEWOOD ELECTRONICS	38
ESR ELECTRONIC COMPONENTS	38
HAMMOND ELECTRONICS Ltd	8
JPG ELECTRONICS	80
MICROCHIP	Cover (ii), Cover (iii), 5
PCBWay	9, 38
PEAK ELECTRONIC DESIGN	Cover (iv)
PICO TECHNOLOGY	11
POLABS D.O.O.	31
QUASAR ELECTRONICS	2
SOUNDTRONICS	3
STEWART OF READING	22
TAG-CONNECT	29
TECHNOBOTS	43

Advertisement offices

Wimborne Publishing Ltd
113 Lynwood Drive
Merley
Wimborne,
Dorset BH21 1UU

Tel 01202 880299
Fax 01202 843233
Email stewart.kearn@wimborne.co.uk
Web www.epemag.com

For editorial contact details see page 7.

Next Month – in the November issue

4 433 619Hz
MHz
kHz
OK
CANCEL
GPS-SYNCED PROGRAMMABLE FREQUENCY REFERENCE
SILICON CHIP
TINNITUS & INSOMNIA KILLER
White
Pink

Tinnitus and Insomnia Killer
Suffer from tinnitus or insomnia? No promises! – but this clever device produces pink or white noise which is widely recognised as easing those problems.

GPS-synced Frequency Reference – Part 2
Next month, we'll walk you through the detailed assembly instructions and provide further information on how to use it, including all the various customisable settings.

Electronic Compass modules
Lean how to use the Elecrow GY-271 and the GY-511 low-cost electronic compass modules. Both incorporate a 3-axis magnetometer and work with an Arduino, Micromite or any other microcontroller which supports I^{2}C.

Using Stepper Motors
In Part 2, we explain how to choose a stepper motor and provide techniques for identifying the leads on surplus motors saved from defunct equipment.

MAXIMITE
Colour Maximite

The Colour Maximite
Whether you want to control external hardware, learn to code, or play a classic arcade game, this retro project is definitely for you. The Colour Maximite is a modern take on an 80s home computer and will provide you with hours of fun.

PLUS!
All your favourite regular columns from *Cool Beans*, *Circuit Surgery and PIC n' Mix, to Electronic Building Blocks*, *Techno Talk* and *Net Work*.

On sale 3 October 2019

Content may be subject to change

WELCOME TO JPG ELECTRONICS
Selling Electronics in Chesterfield for 29 Years

Open Monday to Friday 9am to 5:30pm
And Saturday 9:30am to 5pm

- Aerials, Satellite Dishes & LCD Brackets
- Audio Adaptors, Connectors & Leads
- BT, Broadband, Network & USB Leads
- Computer Memory, Hard Drives & Parts
- DJ Equipment, Lighting & Supplies
- Extensive Electronic Components
  - *ICs, Project Boxes, Relays & Resistors*
- Raspberry Pi & Arduino Products
- Replacement Laptop Power Supplies
- Batteries, Fuses, Glue, Tools & Lots more...

T: 01246 211 202
E: sales@jpgelectronics.com
JPG Electronics, Shaw's Row,
Old Road, Chesterfield, S40 2RB
W: www.jpgelectronics.com

JPG Electronics

Retail & Trade Welcome • Free Parking • Google St View Tour: S40 2RB

Calling all subscribers!

Practical Electronics

If you are one of our valued subscribers then please note that we have changed the way we send subscription renewal reminders.

Instead of sending you a renewal card, we now print a box on the address sheet that comes with your copy of Practical Electronics.

This box will advise you of the last issue of your current subscription.

To renew, you have three easy choices:

1. Call us on: 01202 880299
2. Visit our website at: www.epemag.com
3. Send a cheque with your details to:

Electron Publishing Limited
113 Lynwood Drive
Wimborne
Dorset BH21 1UU

Published on approximately the first Thursday of each month by Electron Publishing Limited, 1 Buckingham Road, Brighton, East Sussex BN1 3RA. Printed in England by Acorn Web Offset Ltd., Normanton WF6 1TW. Distributed by Seymour, 86 Newman St., London W1T 3EX. Subscriptions UK: £25.00 (6 months); £47.00 (12 months); £89.00 (2 years). EUROPE: airmail service, £30.00 (6 months); £56.00 (12 months); £107.00 (2 years). REST OF THE WORLD: airmail service, £37.00 (6 months); £70.00 (12 months); £135.00 (2 years). Payments payable to 'Practical Electronics', Electron Publishing Limited, 113 Lynwood Drive, Wimborne, Dorset BH21 1UU. Email: stewart.kearn@epemag.wimborne.co.uk. PRACTICAL ELECTRONICS is sold subject to the following conditions, namely that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price shown on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter whatsoever.

Faster Programming, Wider Voltage Ranges and Enhanced Interface Options

PICkit™ 4 enables faster design development

MPLAB®
PICkit™ 4
In-Circuit Debugger
Microchip

With five times faster programming and a wider 1.2V to 5V range, the Microchip MPLAB® PICkit™ 4 Development Tool supports low-cost development of even more Microchip controllers.

PICkit™ 4 also introduces improved USB connectivity with a hi-speed USB 2.0 interface and advanced debugging over 4-wire JTAG and serial wire debug with streaming data gateway in addition to legacy interfaces.

The new advanced interfaces support in-circuit programming and debugging of Microchip's CEC1702 hardware cryptography-enabled MCUs, in addition to dsPIC® Digital Signal Controllers and PIC® microcontrollers from 8- to 32-bit.

Microchip

www.microchip.com/PICkit4eu

The Microchip name and logo, the Microchip logo, MPLAB, PIC and dsPIC are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. PICkit and In-Circuit Serial Programming (ISCP) are trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All other trademarks mentioned herein are the property of their respective companies. © 2018 Microchip Technology Inc. All rights reserved.
DS-50002745A. MEC2206Eng04/18

PEAK®
electronic design ltd

Celebrating the return of Practical Electronics!
Limited time offer only

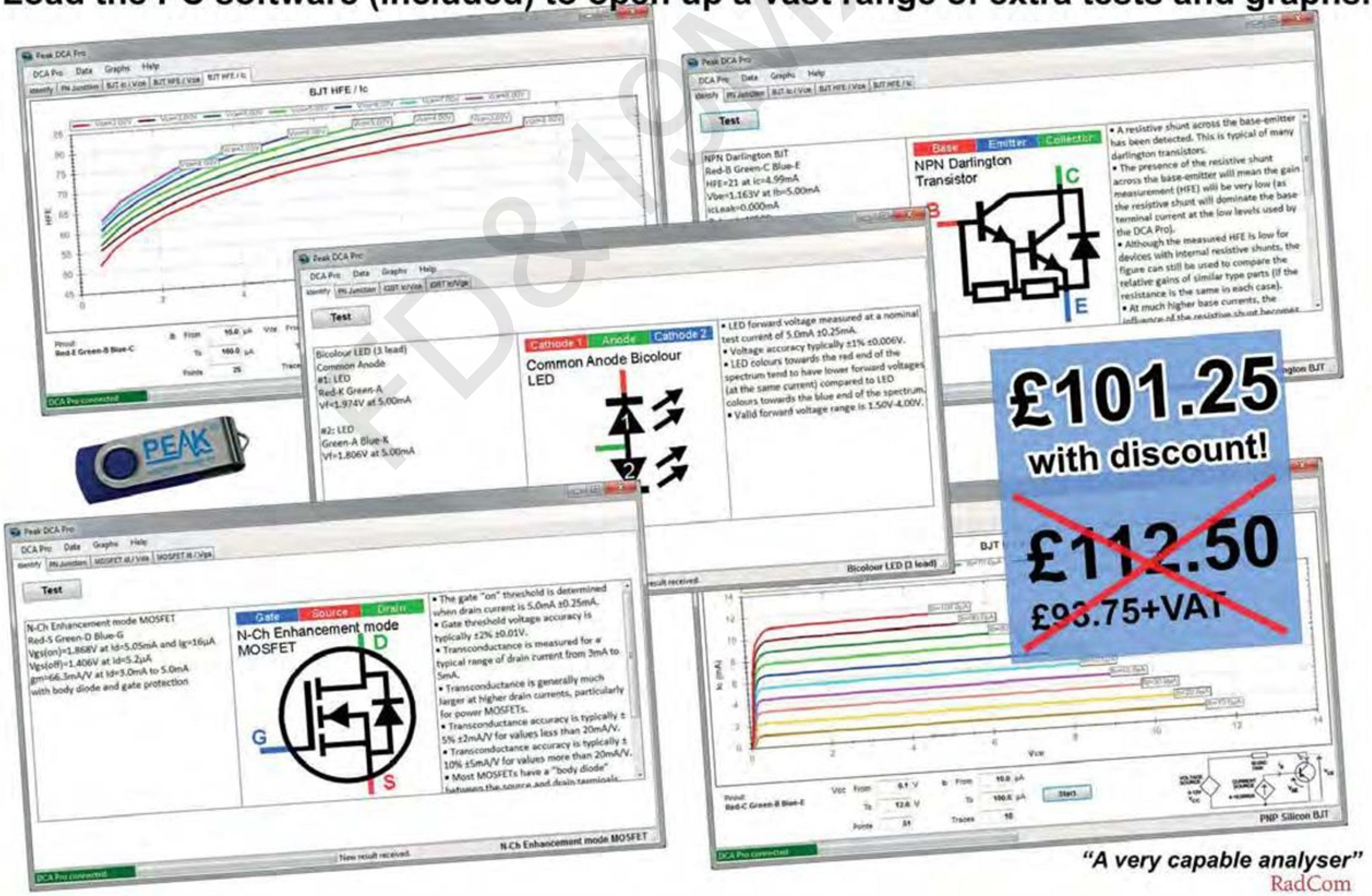
Use code PE10 during checkout
(or on the phone!)

DCA75 Advanced Semiconductor Analyser and Curve Tracer

atlas DCA Pro
DCA75
USB + Graphics
Vbe=1.216V
at Ib=3.6mA
IcLeak=0.000mA
PEAK
electronic design ltd
on - test
scroll - off
warning: avoid charge/voltages

A versatile instrument that you'll love using for testing a huge range of semiconductor components.
Supports MOSFETs, transistors, diodes, LEDs, JFETs, IGBTs and more.
Connect any way round and let the ***DCA Pro*** identify the component type, pinout and lots of parameters.
Can be used standalone or connected to a Windows® PC (XP up to Win10).

Load the PC software (included) to open up a vast range of extra tests and graphs.


Peak DCA Pro
DCA Pro
Data
Graphs
Help
Identify
PN Junction
BJT Ic / Vce
BJT HFE / Vce
BJT HFE / Ic
IGBT Ic/Vce
IGBT Ic/Vge
MOSFET Id / Vds
MOSFET Id / Vgs
Test
BJT HFE / Ic
HFE
Base
Emitter
Collector
B
C
E
Cathode 1
Anode
Cathode 2
1
2
Gate
Source
Drain
G
D
S
Ic (mA)
Vce
PNP Silicon BJT
Bicolour LED (3 lead)
N-Ch Enhancement mode MOSFET
New result received.
DCA Pro connected
Start
PEAK

NPN Darlington BJT
Red-B Green-C Blue-E
HFE=21 at Ic=4.99mA
Vbe=1.163V at Ib=5.00mA
IcLeak=0.000mA

NPN Darlington Transistor

- A resistive shunt across the base-emitter has been detected. This is typical of many darlington transistors.
- The presence of the resistive shunt across the base-emitter will mean the gain measurement (HFE) will be very low (as the resistive shunt will dominate the base terminal current at the low levels used by the DCA Pro).
- Although the measured HFE is low for devices with internal resistive shunts, the figure can still be used to compare the relative gains of similar type parts (if the resistance is the same in each case).
- At much higher base currents, the

Bicolour LED (3 lead)
Common Anode
#1: LED
Red-K Green-A
Vf=1.974V at 5.00mA

#2: LED
Green-A Blue-K
Vf=1.806V at 5.00mA

Common Anode Bicolour LED

- LED forward voltage measured at a nominal test current of 5.0mA ±0.25mA.
- Voltage accuracy typically ±1% ±0.006V.
- LED colours towards the red end of the spectrum tend to have lower forward voltages (at the same current) compared to LED colours towards the blue end of the spectrum.
- Valid forward voltage range is 1.50V-4.00V.

N-Ch Enhancement mode MOSFET
Red-S Green-D Blue-G
Vgs(on)=1.868V at Id=5.05mA and Ig=16µA
Vgs(off)=1.406V at Id=5.2µA
gm=66.3mA/V at Id=3.0mA to 5.0mA
with body diode and gate protection

N-Ch Enhancement mode MOSFET

- The gate "on" threshold is determined when drain current is 5.0mA ±0.25mA.
- Gate threshold voltage accuracy is typically ±2% ±0.01V.
- Transconductance is measured for a typical range of drain current from 3mA to 5mA.
- Transconductance is generally much larger at higher drain currents, particularly for power MOSFETs.
- Transconductance accuracy is typically ± 5% ±2mA/V for values less than 20mA/V.
- Transconductance accuracy is typically ± 10% ±5mA/V for values more than 20mA/V.
- Most MOSFETs have a "body diode"

£101.25
with discount!
~~£112.50~~
~~£93.75+VAT~~

"A very capable analyser"
RadCom

It's only possible to show summary specifications here. Please ask if you'd like detailed data. Further information is also available on our website. Product price refunded if you're not happy.

Tel. 01298 70012
www.peakelec.co.uk
sales@peakelec.co.uk

Atlas House, 2 Kiln Lane
Harpur Hill Business Park
Buxton, Derbyshire
SK17 9JL, UK

UK designed **and** manufactured

FD&19MX7